A deformable particle-in-cell method for advective transport in geodynamic modelling

Henri Samuel

Geophysical Journal International

Geophys. J. Int. (2018) **214,** 1744–1773
Advance Access publication 2018 June 12
GJI Geodynamics and tectonics

doi: 10.1093/gji/ggy231

A deformable particle-in-cell method for advective transport in geodynamic modelling

Henri Samuel[1,2]

[1]*Institut de Physique du Globe de Paris, Sorbonne Paris Cité, CNRS, Paris, F-75005, France. E-mail: samuel@ipgp.fr*
[2]*Institut de Recherche en Astrophysique et Planétologie, Toulouse, F-31400, France*

Accepted 2018 June 11. Received 2018 March 29; in original form 2018 May 23

SUMMARY

This paper presents an improvement of the particle-in-cell (PIC) method commonly used in geodynamic modelling for solving pure advection of sharply varying fields. Standard PIC approaches use particle kernels to transfer the information carried by the Lagrangian particles to/from the Eulerian grid. These kernels are generally 1-D and non-evolutive, which leads to the development of under- and oversampling of the spatial domain by the particles. This reduces the accuracy of the solution, and may require the use of a prohibitive amount of particles in order to maintain the solution accuracy to an acceptable level. The new proposed approach relies on the use of deformable kernels that account for the strain history in the vicinity of particles. It results in a significant improvement of the spatial sampling by the particles, leading to a much higher accuracy of the numerical solution, for a reasonable computational extra cost. Various 2-D tests were conducted to compare the performances of the deformable PIC (DPIC) method with the PIC approach. These consistently show that at comparable accuracy, the DPIC method was found to be four to six times more efficient than standard PIC approaches. The method could be adapted to 3-D space and generalized to cases including motionless transport.

Key words: Numerical modelling; Numerical solutions; Dynamics: convection currents, and mantle plumes; Dynamics of lithosphere and mantle.

1 INTRODUCTION

The accurate modelling of the advection of compositional heterogeneities is a common requirement for computational geodynamics at scales ranging from planet-size (Gerya & Yuen 2007; Lin *et al.* 2011), mantle (Hoïnk *et al.* 2005; McNamara & Zhong 2005; Samuel & Bercovici 2006; Tackley 2008; Maurice *et al.* 2017), core (Bouffard *et al.* 2017), lithospheric scale (Poliakov & Podladchikov 1992; van Hunen *et al.* 2004; Gerya 2010; and references therein), down to the scale of magma chambers of just a few tens of metres thick (Ruprecht *et al.* 2008).

These experiments rely on the modelling of a purely advective transport equation written below for a vector quantity **C** in a flow field **u**:

$$\frac{\partial \mathbf{C}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{C} = 0, \tag{1}$$

where t is the time. The above equation is relevant to cases where **C** represents active or passive compositional fields being advected, and for which diffusion can be neglected.

In the frame of geodynamic modelling, the flow field **u** generally corresponds to the solution of the Navier–Stokes equations. The latter is generally obtained via the discretization of these equations on Eulerian grids, which are often coupled to supplementary equations for the conservation of additional quantities such as internal energy (e.g. Hoïnk *et al.* 2005; McNamara & Zhong 2005; Tackley 2008; Bouffard *et al.* 2017; Maurice *et al.* 2017).

The presence of Eulerian grids to compute the flow field **u** and other quantities makes the use of the 'particle-in-cell' (PIC) method for solving eq. (1) suitable and advantageous in geodynamic modelling. This hybrid approach combines an Eulerian description of the field **C** with a network of Lagrangian particles moving through the grid. In the original version of the method (Harlow 1957, 1964), particles carried a limited amount of information: identity and mass. The PIC method was later adapted to a broader range of applications in various fields of research that led to additional quantities to be carried by the particles (e.g. energy, temperature, momentum...).

Regardless of its countless evolutions, the PIC method aims at making use of the best of both worlds: while processes such as diffusion can generally be accurately and straightforwardly computed on an Eulerian grid, the evaluation of the advective transport term using a Lagrangian formalism can be advantageous because it does not involve by itself significant amounts of numerical dissipation compared to Eulerian approaches (Harlow 1957; Monaghan 1985; Rider & Kothe 1995; Kothe 1998).

To some extent, the PIC approach can be viewed as an operator splitting technique where at each time step motionless processes are updated on the Eulerian grid, and transferred to the particles.

Then, advection is handled via the Lagrangian particles prior to the mapping of the advected quantities back onto the Eulerian grid. Therefore, the method requires particle–mesh and mesh–particle mappings to be specified. These critical operations constitute a major source of inaccuracy in the PIC solution (Monaghan 1985; Deubelbeiss & Kaus 2008; Duretz *et al.* 2011; Thielmann *et al.* 2014). Indeed, while the Lagrangian advection alone is not prone to significant numerical diffusion, particle–mesh mappings can introduce important amounts of dissipation. This is particularly true when the spatial distribution of particles is not homogeneous, leading to areas in the vicinity of gridpoints that are not sufficiently well sampled by particles, and other regions where the domain is oversampled by particles. This recurrent sampling problem develops in regions characterized by strong deformation, and concerns both compressible and incompressible flow (Wang *et al.* 2015; Pusok *et al.* 2016). The non-homogeneous sampling has two main origins. The first one corresponds to inaccuracies in advecting the Lagrangian particles (Meyer & Jenny 2004). This aspect has drawn the attention of a few recent studies (Wang *et al.* 2015; Pusok *et al.* 2016), which have proposed the use of conservative schemes to map velocity components from the Eulerian grid to the Lagrangian particles during their advection. Such schemes have shown to significantly improve the accuracy of the interpolation, and result in a considerably more homogeneous spatial sampling. The second origin, which has received less attention, is related to the deforming nature of the flow (e.g. Moresi *et al.* 2003), and is completely independent of the accuracy of the numerical methods for interpolating the velocities at particles' locations. In fact, for a given velocity field, particles should travel along their characteristics, and even in the case of incompressible flows, the distance between characteristics can vary in general, and can strongly diverge or converge in regions characterized by strong deformation. This naturally leads to the development of a non-homogeneous spatial distribution of the Lagrangian particles, even if the particles locations are perfectly known, as we shall see later.

In addition to the use of better interpolation and time-integration schemes, common remedies to the aforementioned non-homogeneous spatial sampling are (Poliakov & Podladchikov 1992; van Keken *et al.* 1997; Tackley & King 2003; Gerya 2010; Wang *et al.* 2015; and references therein) (1) the increase of the number of Lagrangian particles, which can lead to a prohibitive computational cost; (2) the redistribution of particles via seeding and deletion of particles in under- and oversampled regions, respectively. Besides the extra cost associated with these operations, particle remeshing introduces significant amounts of numerical diffusion in undersampled regions.

This paper therefore focuses on the resolution of eq. (1) with an improved version of the PIC method, which yields a better behaviour in regions characterized by strong deformation. For simplicity, I consider exclusively the case of purely advective transport in 2-D Cartesian domains, although PIC methods can be applied when advective transport is coupled to other processes such as diffusion (Brackbill *et al.* 1987; Gerya & Yuen 2003) and generalized to 3-D geometry.

The new approach I propose for the PIC method is based on the use of anisotropic particle 'kernels' (i.e. the representation of their spatial domain of influence) that account for the deformation history in the vicinity of particles. This new method hereafter termed 'Deformable PIC' (DPIC), yields a significant improvement of the domain sampling by particles, without requiring a prohibitive extra computational cost.

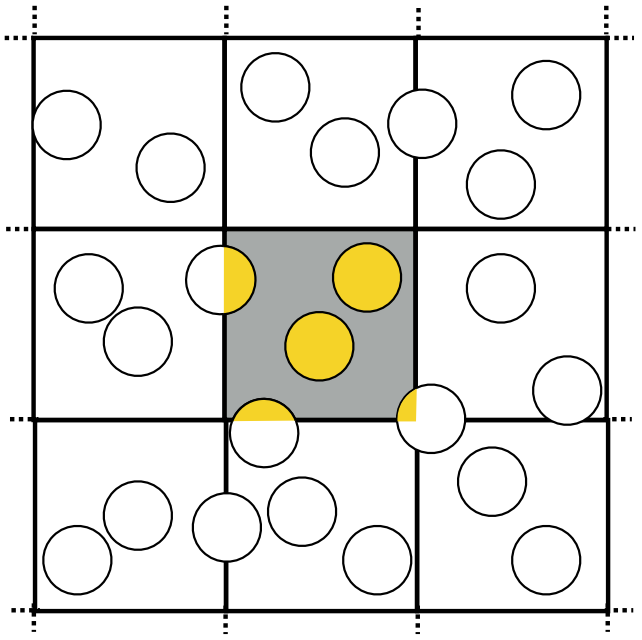

Figure 1. Schematic representation of the particle kernels and grid shape function S in a domain discretized with square cells of dimensions $h \times h$. In this example, the grid shape functions take the form of the NGC (eq. 8), interacting with disc particle kernels of identical radius r_p (eq. 11). Focusing on the central cell located at $\mathbf{x}_c$, the grey area corresponds to $S(\mathbf{x}_c) = 1$, and $S(\mathbf{x}_c) = 0$ elsewhere. The boundaries of the particle kernels are shown in black. The common area between each particle kernels and $S(\mathbf{x}_c)$ are shown in yellow. Therefore, $W(\mathbf{x}_c)$ is the sum of areas shown in yellow normalized by the cell volume, h^2 (eqs 6 and 7). See text for further details.

The paper is structured as follows: Section 2 summarizes the standard implementation of the PIC method and illustrates its limitations. Section 3 presents the new particle kernel used in the DPIC method. Section 4 describes the implementation details of the DPIC method. Section 5 illustrates the benefits of the new method. Section 6 compares the performances of the standard and the DPIC methods using kinematic and dynamic flow tests. Section 7 summarizes the conclusions and proposes possible future developments, and applications of the DPIC formalism.

2 THE PIC METHOD

As in purely Eulerian methods the physical domain, Ω, is discretized using a finite number of points/control volumes. For simplicity, throughout this study we adopt a 2-D Cartesian framework $\mathbf{x} = (x, z)$, where the domain is discretized on a half-staggered grid using $n_x \times n_z$ square cells of size h^2, whose centres are located at $\mathbf{x}_c$. Therefore, $\mathbf{C}(\mathbf{x}_c) = \mathbf{C}(x_c, z_c) \equiv \mathbf{C}(i, j)$, where the grid indexes i and j point to the coordinates $x_c = h(i - 0.5)$ and $z_c = h(j - 0.5)$. Velocity components are specified at cell vertices. The choice for such grid configuration is purely arbitrary and is unlikely to have a particular effect on the results presented.

To complete the previous Eulerian description, a set of n_p Lagrangian particles is used to describe the vector field $\mathbf{C}$ in eq. (1). Each particle represents a macroscopic fluid sample/parcel at a given location $\mathbf{x}_p$, which is determined via the integration of the following ordinary differential equation:

$$\frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} = \mathbf{u}_p, \tag{2}$$

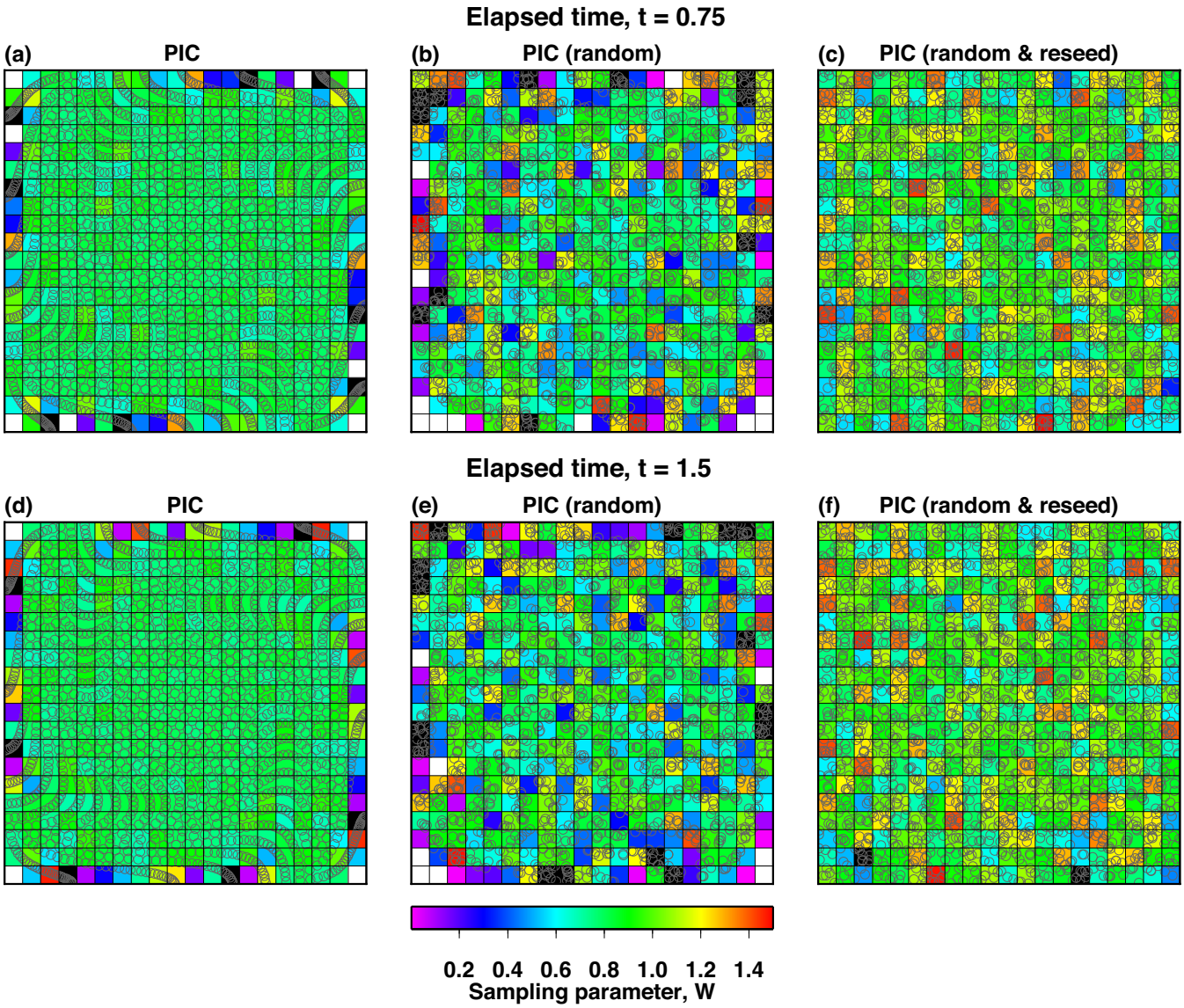

Figure 2. Results of the vortex flow test at $t = 0.75$ (top) and $t = 1.5$ (bottom), obtained using the PIC method with initially four particles per cell. The domain is discretized using 20×20 square cells. The cell sampling parameter W is displayed in colour. White and black indicate cells that are unsampled ($W = 0$) and strongly oversampled ($W > 1.5$) by particle kernels, respectively. The contour of particles kernels is shown in black. Left-hand, middle and right-hand panels correspond to homogeneous and regular initial positions of particles with no remeshing, randomly perturbed initial positions of particles with no remeshing, randomly perturbed initial positions of particles with particle remeshing applied in under- and oversampled areas, respectively. See text for further details.

where $\mathbf{u}_p = \mathbf{u}(\mathbf{x}_p)$ is obtained by interpolation of the velocity field at the Eulerian grid points. Eq. (2) may be integrated using a Total Variation Diminishing–Runge–Kutta (TVD–RK) scheme of second or third order (Shu & Osher 1988).

Each particle carries information $\mathbf{C}_p$ obtained from the interpolation of neighbouring Eulerian field values. These Eulerian-to-Lagrangian interpolations are performed by applying weights $w_g(\mathbf{x}_p)$ to the nearest grid cell (NGC) to which each particle belongs:

$$\mathbf{C}_p = \sum_{g=1}^{n_{\mathrm{NGC}}} w_g(\mathbf{x}_p)\, \mathbf{C}(\mathbf{x}_g), \tag{3}$$

where $n_{\mathrm{NGC}} = 4$ in 2-D and 8 in 3-D.

In this study we will restrict to the use of linear distance weights, w_g, applied to the four NGC surrounding each particle. This avoids unphysical undershoots and overshoots upon interpolation when dealing with sharply varying fields (Monaghan 1985).

Similarly, the Eulerian field is expressed using a weighted arithmetic mean:

$$\mathbf{C}(\mathbf{x}_c) = \sum_{p=1}^{n_p} w_p\, \mathbf{C}_p, \tag{4}$$

where the particle weights, w_p, are expressed as

$$w_p = \frac{w_*(\mathbf{x}_c, \mathbf{x}_p)}{W(\mathbf{x}_c)}. \tag{5}$$

The area-based weight, w_*, corresponds to a convolution product (Brackbill 2005):

$$w_*(\mathbf{x}_c, \mathbf{x}_p) = \frac{S(\mathbf{x}_c) * K_p(\mathbf{x}_p)}{h^2}, \tag{6}$$

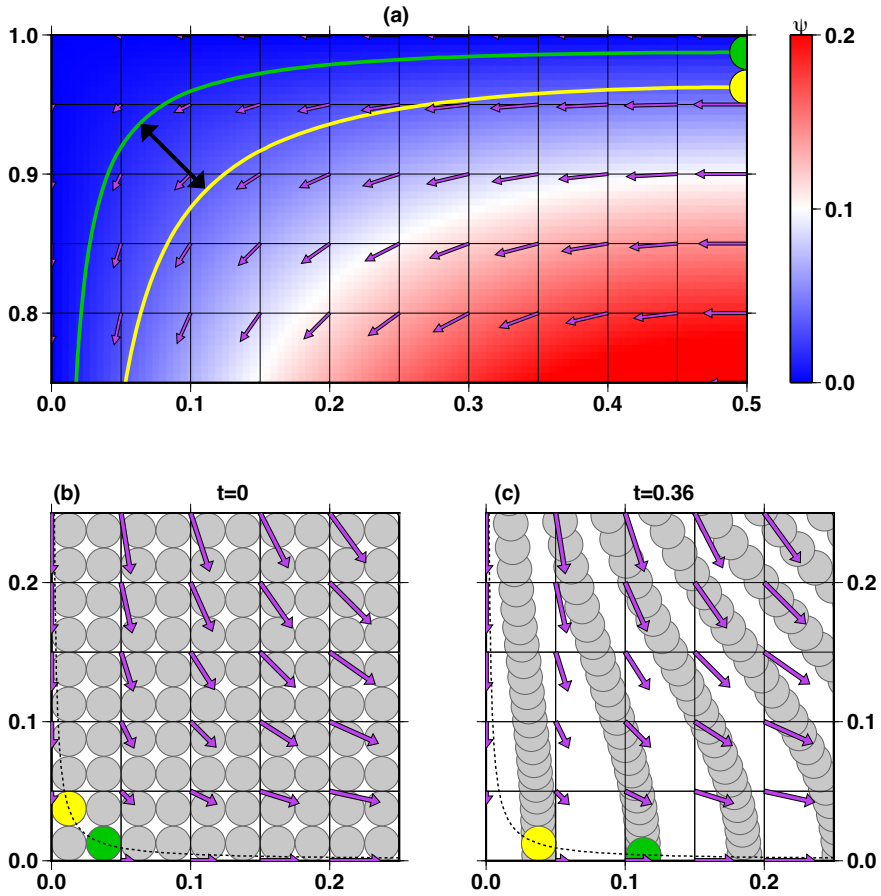

Figure 3. Results of the vortex flow test obtained using the PIC method with initially four particles per cell. The domain is discretized using 20 × 20 square cells. (a) Close-up view of the model domain showing the trajectories for two particles, A (green) and B (yellow) initially close to each other. The imposed streamfunction field, ψ, is shown. The black arrow shows the shortest distance between the trajectories, which is maximum at each corner of the domain, and becomes larger than a cell diagonal, causing non-homogeneous sampling of the domain upon advection. (b and c) Advection of disc particle kernels in the vicinity of $x = z = 0$. Particle C (yellow) and D (green) that belong to the same streamline/trajectory (dotted line) are shown. The pure-shear deformation in this region results in an exponential separation of C and D. The gap produced is not filled out by other nearby particles, which follow similar parallel trajectories, leaving cells empty. The velocity vectors are shown by the purple arrows. See text for further details.

where the shape function S defines the region in the vicinity of a grid cell centre where the value of $\mathbf{C}$ can be influenced by particles, whose domains of influence are expressed by the kernel K_p. Therefore, $w_*(\mathbf{x}_c, \mathbf{x}_p)$ represents the area common to S and K_p normalized by the grid cell area, as illustrated in Fig. 1. As described in the following, both S and K_p are also functions of $\mathbf{x}$, but to simplify the notation this dependence is not explicitly written.

To ensure that the averaging scheme is convex, the particles' weights are normalized by a cell sampling parameter:

$$W(\mathbf{x}_c) = \sum_{p=1}^{n_p} w_*(\mathbf{x}_c, \mathbf{x}_p). \tag{7}$$

This quantity measures the volume fraction of a given cell sampled by particles. $W(\mathbf{x}_c) > 1$ indicates an oversampled cell, while $W(\mathbf{x}_c) < 1$ reflects an undersampled cell. Note that since W ignores particle kernel overlaps, this sampling parameter should be viewed as a proxy for the quality of sampling. However, not removing overlapping areas between particle kernels is preferable as it would imply a mass/volume loss.

2.1 Grid shape functions

Several possibilities exist for the choice of S. A common form used in this study and elsewhere (e.g. Samuel & Farnetani 2003; Tackley & King 2003; Thielmann *et al.* 2014) is the NGC function, expressed here in 2-D (x, z) space:

$$S(\mathbf{x}_c) = s(h/2, |x - x_c|)\, s(h/2, |z - z_c|), \tag{8}$$

with

$$s(r, d) = \begin{cases} 1 \text{ if } d \le r \\ 0 \text{ if } d > r. \end{cases} \tag{9}$$

This results in $S = 1$ inside a grid cell of size $h \times h$, and $S = 0$ elsewhere. Alternatively, the grid cell shape function can be set to

$$S(\mathbf{x}_c) = s(h, |\mathbf{x} - \mathbf{x}_c|). \tag{10}$$

In this case, S will be one in a disc of radius $|\mathbf{x} - \mathbf{x}_c|$ centred on $\mathbf{x}_c$, and zero elsewhere (Fig. 1). Smoother versions of S are also commonly considered by replacing $s(h, |\mathbf{x} - \mathbf{x}_c|)$ in the above expressions by linear truncated functions (Gerya & Yuen 2003; Tackley & King 2003; Duretz *et al.* 2011), Gaussian functions (Brackbill 2005) or piecewise high-order polynomials such as splines (Monaghan 1985, 1992).

2.2 Particle kernels

Particle kernels can take forms similar to grid cell shape functions:

$$K_p(\mathbf{x}_p) = s(r_p, |\mathbf{x} - \mathbf{x}_p|), \tag{11}$$

where r_p expresses the particle's spreading. If r_p is a constant, the above kernel corresponds to a disc/sphere of radius r_p centred around the particle (Fig. 1).

A more common form used in geodynamic modelling is the point-wise kernel: $K_p(\mathbf{x}_p) = \delta(|\mathbf{x} - \mathbf{x}_p|)$ (Gerya & Yuen 2003; Tackley & King 2003; Deubelbeiss & Kaus 2008), where δ is the Dirac distribution. On the contrary, disc-shaped particle kernels (eq. 11, Fig. 1) are rarely used in geodynamic modelling, because they require surface/volume integration. This is more complicated to implement, and computationally more expensive to use than point-wise kernels. However, despite this handicap, I will refer to them in the first set of experiments presented in this paper, because of their conceptual simplicity, their generality (i.e. they include point-wise kernels) and their physical sounding [disc/spheres are intuitively more easily associated with macroscopic area/volume or fluid than material points, which is consistent with Harlow's original proposal for the PIC method (Harlow 1957, 1964)].

Note that particle kernels do not need to be identical to each other. In addition, K_p can evolve with time, as is the case for adaptive particle kernels (see Monaghan 1985 and references therein).

With the exception of point-wise particles, the kernel functions explicitly involve a characteristic length scale, r_p, whose appropriate value may be problem-dependent. Nevertheless, since the quantities carried by the particles result from the interpolation of the Eulerian grid values, it is desirable to have r_p smaller than the Eulerian grid spacing h, which could yield, to some extent, subgrid resolution, depending on the smoothness of $\mathbf{C}$ (Grigoryev *et al.* 2002; Brackbill 2005). However, a value of r_p too small would involve a prohibitively large number of particles. This remains true for the case of point-wise kernels, where particles have a statistical meaning rather than a macroscopic meaning, which still requires a significant number of particles to yield a representative description. Another fundamental remark is that the kernel functions commonly used in the PIC method are 1-D 'spherically' symmetric.

Ideally, for computational efficiency and accuracy, one seeks a configuration that provides the best sampling of the domain Ω by the particles at all times, at the lowest computational cost. Namely, assuming that cell shape functions take the form of eq. (8), one seeks a set of $K(\mathbf{x}_p)$ such that, ideally, $W(\mathbf{x}_c) = 1 \,\forall\, \mathbf{x}_c \in \Omega$, that is, the volume fraction sampled by particles present in each computational cell equals the cell volume. For point-wise particles, the corresponding requirement would translate into a Voronoi-type statement that the average distance between nearest particles remains constant. Sets of particles that are too far away from the above requirement tend to result in over- or underrepresentations of the field $\mathbf{C}$, which are computationally inefficient or inaccurate, respectively. Approaching the above requirement implies an infinite number of particles, or complex irregular kernel functions, both leading to a prohibitive computational cost. A somewhat looser requirement could therefore be that 'most' (e.g. ∼75 per cent) of each grid cell volume are sampled by particles:

$$W(\mathbf{x}_c) = 0.75 \,\forall\, \mathbf{x}_c \in \Omega. \tag{12}$$

While the value of 0.75 was chosen arbitrarily, tests conducted indicate when this empirical requirement is met, the quality of spatial sampling and the solution accuracy were found to be satisfactory. In 2-D space, this corresponds approximately to n^2 identical particles with tangent disc kernels of radius $r_p = h/(4n)$, within a square cell of volume h^2 (see for instance Fig. 5a).

For point-wise kernels, it is more convenient to use a normalized version of the sampling parameter:

$$W^*(\mathbf{x}_c) = W(\mathbf{x}_c)/W_0, \tag{13}$$

where W_0 corresponds to the average cell sampling value at initial time. If one assumes that the volume associated with each particle is identical, the above definition implies that $W^* = n_{pc}(\mathbf{x}_c)/n_{pc0}$, where n_{pc0} is the average initial number of particles per grid cell, and $n_{pc}(\mathbf{x}_c)$ is the number of particles in a given cell whose centre is $\mathbf{x}_c$. Similar to the bulk cell sampling, W, W^* measures the relative sampling within a given cell by particles. $W^*(\mathbf{x}_c) > 1$ indicates an oversampled cell, while $W^*(\mathbf{x}_c) < 1$ reflects an undersampled cell, and the requirement stated in eq. (12) would correspond to $W^* = 1$. The normalized sampling parameter is useful to compare samplings between point-wise and more general (e.g. disc) kernels.

Overall, the values of W or W^* directly impact the accuracy and the efficiency of PIC methods, which rely heavily on the choice of an appropriate kernel function, K_p, that connects the Eulerian grid with the Lagrangian network formed by the set of particles.

2.3 Basic algorithm for the PIC method

In the frame of pure advection considered here, the PIC method is implemented as follows: Particle positions are initially regularly distributed over the domain, by prescribing a fixed number of particles per grid cell, n_{pc0}, leading to a total initial number of particles $n_{p0} = n_{pc0}n_xn_z$. In order to reduce the development of sampling problems by the particles, their initial position may also be randomly assigned within the domain. The quantities $\mathbf{C}_p$ carried by each particle are either initially explicitly specified, or can be determined from the interpolation of the neighbouring grid values (eq. 3). Next, the integration of eq. (1) results in the following algorithm applied at each time step:

(1) Particles advection: this step is performed by numerically solving eq. (2) using a TVD–RK scheme of second order (unless specified otherwise) for each particle, using a Courant–Friedrich–Lewy based (CFL) time step, namely $0.25h/(\max(|u_x|) + \max(|u_z|))$. Note that during this stage, the interpolation of the velocities from the grid to the particles is performed using the conservative scheme of Meyer and Jenny (2004).

(2) Particle remeshing: the number of particles present in each grid cell is evaluated in order to detect and to correct for possible over- or undersampled areas. If the number of particles present in a given cell exceeds a threshold fixed at $2\,n_{pc0}$, particles are removed. On the other hand, if a grid cell is found to be empty, n_{pc0} new particles are randomly seeded (within the desired cell), and their associated field values $\mathbf{C}_p$ are assigned using mesh–particle mapping (eq. 3). Other criteria were considered and tested but did not significantly alter the results shown in this study.

(3) Conversion of Lagrangian to Eulerian grid values, using particle–mesh mappings (eq. 4).

The particle remeshing stage is optional. In the case where empty cells develop, it could be replaced by the use of a background value, or the use of wider cell shape functions upon particle–mesh mappings (Wang *et al.* 2015 and references therein). In this work, the second option will be favoured when using point-wise kernels. While each of these approaches has its own advantage and inconvenience, both of them introduce dissipation into the solution.

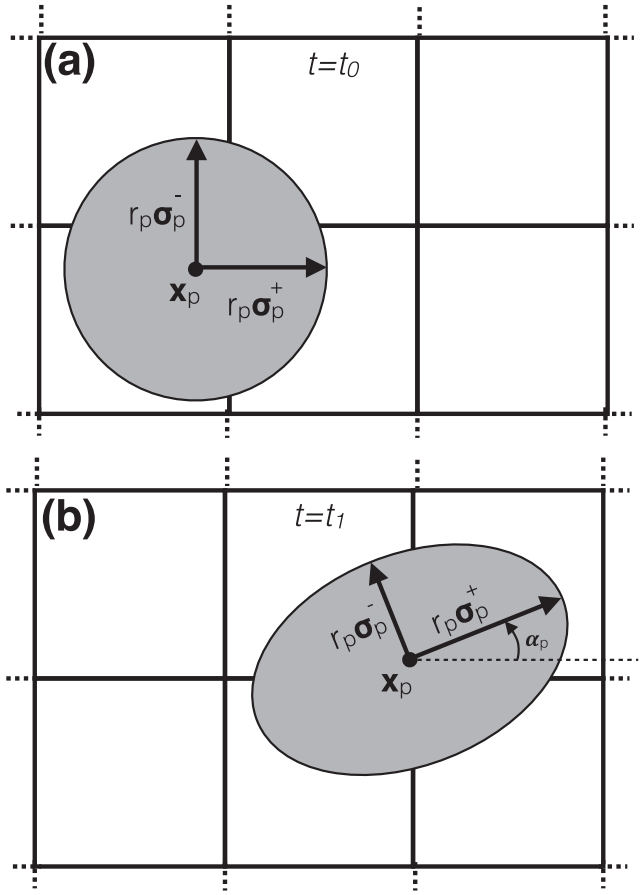

Figure 4. Schematic representation of the time-evolving elliptical kernels for the DPIC method between an initial time t_0 and a following instant t_1. At $t = t_0$, the semi-major and semi-minor axis lengths are identical: $\sigma_p^+ = \sigma_p^- = 1$, and the angle between the major semi-axis and the x-axis is set to 0 ($\alpha_p = 0$). The velocity field, **u**, known at Eulerian grid locations is interpolated in order to advance the location of the ellipse centre $\mathbf{x}_p$ and to update the values of σ_p^+, σ_p^- and α_p. In this example, **u** is assumed to generate shear and rotation. See text for further details.

2.4 Evaluation of the spatial sampling for the PIC method

To illustrate the weaknesses of the PIC method described above, I considered a simple incompressible steady vortex flow field used in van Keken *et al.* (1997) (albeit a constant) and similar to the one used in Rider & Kothe (1995), in which velocity components are derived from the streamfunction expression: $\psi(x, z) = \sin(\pi x)\sin(\pi z)/\pi$. This leads to

$$\mathbf{u} = (\partial_z\psi, -\partial_x\psi)^T = [\sin(\pi x)\cos(\pi z), -\cos(\pi x)\sin(\pi z)]^T, \quad (14)$$

where the exponent T indicates the transpose. Such a velocity field produces both shearing and rotation within Ω, with free-slip boundary conditions along all side walls. The Cartesian domain [0,1]×[0,1] is discretized using 20×20 square cells. This example serves only to illustrate the spatial sampling problems that develop in the case of a simple steady velocity field. Therefore, I strictly focus on the distribution of the particle positions and their corresponding sampling of the spatial domain, and I do not consider the evolution of a field **C**. This is equivalent to considering the trivial case were the field $\mathbf{C} = C_0$ has a constant value and the particles carry the same identity $\mathbf{C}_p = C_0$. I used four particles per grid cell with simple disc kernels: $K_p = s(h/4, |\mathbf{x}_p - \mathbf{x}|)$ (see eq. 11).

Three configurations were considered for the implementation of the PIC method: (1) a regular initial distribution of the particles' positions without remeshing, (2) random initial particle positions without remeshing and (3) random initial particle positions combined with a remeshing procedure described in the previous section.

Fig. 2 shows the contours of the particle kernels at an early stage ($t = 0.75$) and a later stage ($t = 1.5$), for the three cases mentioned above. These snapshots in time also display the values of the sampling parameter W calculated for each cell centre. For a regular distribution of the particles' initial positions, non-homogeneous particle sampling rapidly develops, as can be observed in Figs 2(a) and (d) with the presence of unsampled cells ($W = 0$) coexisting with strongly oversampled cells ($W > 1.5$). These sampling problems initially develop in the vicinity of regions characterized by the largest deformation, which in the present case, correspond to the four corners of the domain, where the flow generates pure shear. As mentioned previously, it is not abnormal that high deformation results locally in particle rarefaction and clustering. This outcome is primarily due to the convergence and the divergence of streamlines in regions of strong deformation, and even with an error-free Lagrangian advection, particle rarefaction and clustering would still naturally develop. As mentioned earlier, particles should travel along their characteristics, that is, the flow streamlines, which in the present case, are known exactly. Consider two particles, named hereafter A and B, that are initially close to each other. Particle A is initially located at $x = x_{A_0} = 0.5$ and $z = z_{A_0} = 1 - r_0$. Particle B is located at $x_{B_0} = x_{A_0}$ and $z_{B_0} = z_{A_0} + 2r_0 = z_{A_0} - h/2$ (Fig. 3a). These differences in initial positions imply that particles A and B sample two different streamlines, represented by distinct values of the streamfunction: $\psi_A = \pi \sin(\pi z_{A_0})$ and $\psi_B = \pi \sin(\pi z_{B_0})$. While particles A and B should ideally remain on their own streamlines at all times, they will travel along them at different speeds such that most of the time they will not align together with the centre of the domain. However, since they evolve on closed trajectories, their minimum spacing, d_{AB} is bounded by the shortest distance between $\psi(x, z) = \psi_A$ and $\psi(x, z) = \psi_B$. The chosen expression for the streamfunction dictates that d_{AB} is maximum in the vicinity of each corners of the domain, when $x_A(t) = z_A(t)$ and $x_B(t) = z_B(t)$. This leads to

$$d_{ABmax} = \frac{\sqrt{2}}{\pi}\left[\arcsin(\sqrt{\pi\psi_B}) - \arcsin(\sqrt{\pi\psi_A})\right]. \quad (15)$$

The above relationship yields $d_{ABmax} \approx 1.4h$, which indicates that the distance between the trajectories of two initially close particles can stretch by a distance comparable to the diagonal of one grid cell, which can cause non-homogeneous sampling of the domain by the particles, and even the development of empty cells (Fig. 3a).

The phenomenon described above is largely amplified by the fact that along a given streamline velocity changes, resulting in possibly large variations in the distances between two consecutive particles. This aspect can be illustrated with the given flow, in the vicinity of any of the four corners of the domain. For convenience I chose the lower left corner (i.e. located at $x = z = 0$). Consider now two particles, C and D, initially located at $x_{C_0} = 0.25h$, $z_{C_0} = 0.75h$ and $x_{D_0} = z_{C_0}$, $z_{D_0} = x_{C_0}$, such that C and D are initially located within the same corner cell, and belong to the same streamline (Fig. 3c). While these two particles follow the same trajectory, the velocity along the corresponding streamline varies strongly (Figs 3b–c) in this region, and so does l_{CD}, the distance between particles C and D. One can show that, in the vicinity of each corner, l_{CD} grows exponentially with time: $l_{CD} > h\exp(\pi t)/2$ (see Appendix A for the derivation of this expression). According to the above expression,

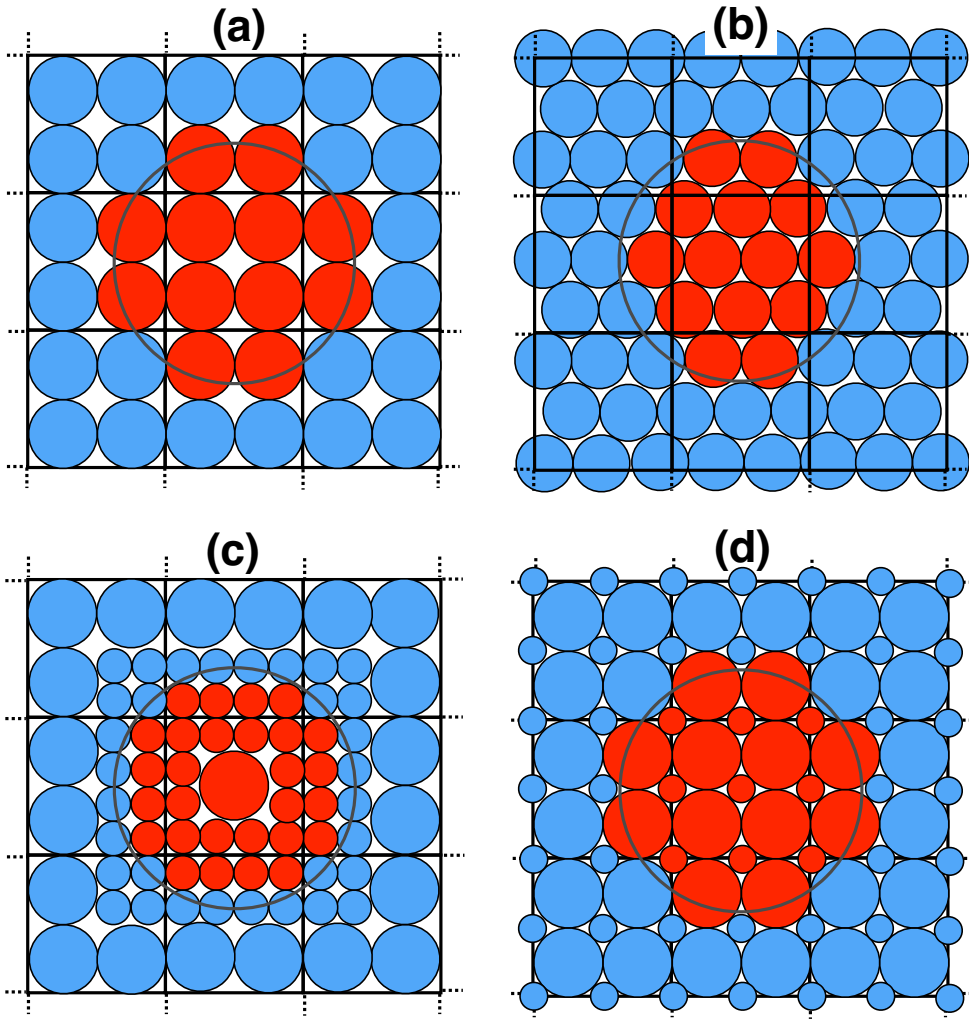

Figure 5. Examples of different initial distributions of Lagrangian particles and their kernels during the initialization stage of the DPIC method. The assignment of particles identities corresponds to the values of $C = 1$ outside or $C = 2$ within the dark grey circle. Particles located within the circle are of type 2 (red), while the rest of the particles population are of type 1 (blue). See text for further details.

the distance between particles C and D becomes larger than 1.5 grid cell when $t = \ln(3)/\pi \approx 0.36$ or more. During this time window, particle D has travelled from (x_{D_0}, z_{D_0}) to $(x \approx 2.2h, z \approx 0.1h)$. That is, during the amount of time necessary for particle D to cross a distance smaller than the size of two grid cells, the spacing between two disc particles kernels of radius $h/4$ initially contained within the same grid cell has grown larger than the size of one cell. The gap formed between the two particles (and their corresponding kernels) is not filled out by other particles, because flow trajectories in the vicinity of C and D are similar (i.e. they essentially follow a translation along the x-direction, see Figs 3b–c). Consequently, the presence of regions with strong deformation results in the creation of gaps between the particles, which can rapidly cause cells to be empty or oversampled. It is important to stress that this occurs even in an incompressible flow, and independently of the accuracy of particles' advection. This is a fact that is not always fully recognized in the literature, where it is sometimes expected that no particle convergence or divergence should occur in incompressible flows. The above experiments demonstrate that this is not the case: particle divergence or convergence is generally consistent with mass conservation in an incompressible flow, except for trivial cases (solid rotation and translation). While processes leading to particle convergence or divergence occur through the presence of specific regions (e.g. stagnation points), such singular points are ubiquitous in non-trivial velocity fields where deformation occurs.

The undesirable effects described above can be reduced by considering an initial random distribution of particles within the domain. The randomness allows a greater number of distinct streamlines/trajectories to be sampled, yielding an observable improvement. However, the benefit remains limited, and both under- and oversampled regions still develop, as seen in Figs 2(b) and (e). A further reduction in particles rarefaction and clustering can be obtained by increasing the number of particles. This also corresponds to an increase in the number of characteristics followed by particles. However, for cases involving pure shear, the presence of singular stagnation points would require a prohibitive amount of particles, even in 2-D. Therefore, a more practical compromise is to perform particle remeshing in under- and oversampled regions. While this efficiently removes particles rarefaction and clustering (Figs 2c and f), these operations lead to a significant (10–20 per cent) computational extra cost, and introduce numerical diffusion, as will be seen in the next sections. This is due to the fact that additional mesh–particle mappings are required in order to determine the values of the quantities carried by the newly added particles. As mentioned

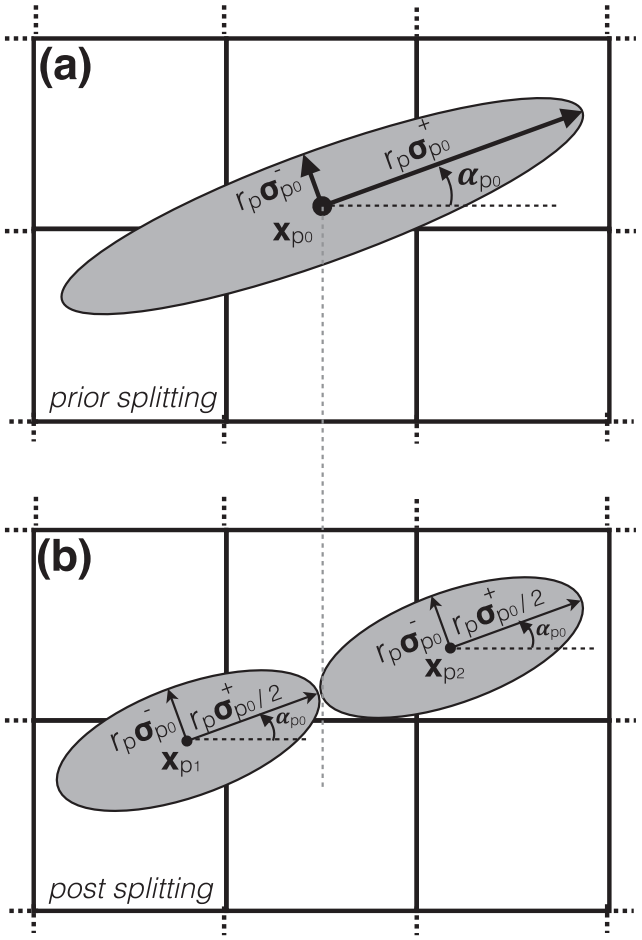

Figure 6. Schematic representation of the particle kernel splitting procedure. See text for further details.

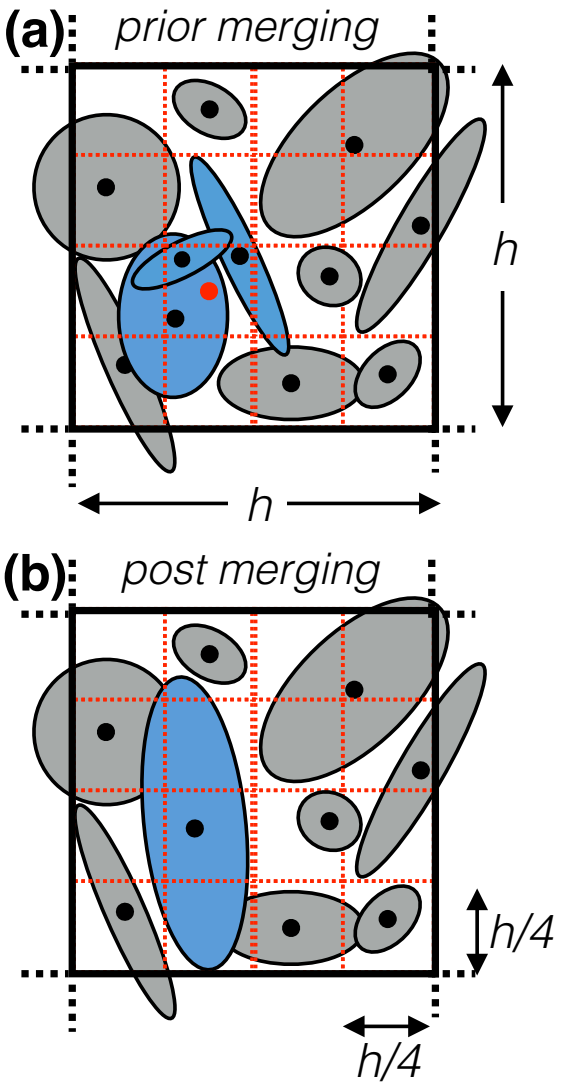

Figure 7. Schematic representation of the particle kernel merging procedure. See text for further details.

previously, such a mapping involves interpolations, thus a potential loss of information. We note that even in this case the spatial sampling by particle kernels is not homogeneous with important fluctuations in W. This also results from both the initial random position of particles, and the simplicity of the remeshing procedure used here. While more efficient sampling could be implemented for the random initialization and the remeshing procedure (e.g. Edwards & Bridson 2012) such that particle kernels overlap would be further reduced, the extra cost involved with such improvement would rapidly become prohibitive. In addition, remeshing procedures involve interpolations. Even with the use of higher-order interpolation schemes, sharp variations in the compositional field tracked by the particles remain problematic and can induce significant amounts of numerical diffusion, along with unphysical behaviour (e.g. Gerya & Yuen 2003 and references therein).

Overall, the example described hereabove illustrates the weakness of the PIC method, which mostly results from the fact that the fixed 1-D particle kernels are not well adapted to deal with convergent and divergent flow streamlines, because standard kernels do not account for the deformation of fluid parcels in the vicinity of the particles. Note that these conclusions would be even more pronounced for point-wise particle kernels more commonly used in geodynamic modelling. Indeed, for example, one can easily see in Figs 2(b) and (e) that the number of cells unsampled by particle kernel centres would be even more important.

3 DEFORMABLE PARTICLE KERNELS

As an alternative to the spherically symmetric kernels commonly used in PIC methods, I propose the use of anisotropic, time-evolving kernels, whose characteristics are determined based on the Lagrangian strain history induced by the velocity field. The basic idea behind these kernels is that the latter, which represent a fluid parcel in the vicinity of particles, should naturally evolve as the result of the deformation imposed by fluid motion, rather than being simply translated, as commonly assumed in most PIC calculations. Such kernels, which will subsequently be referred to as 'deformable particle kernels', constitute the basis of the novel DPIC method presented in this paper.

To derive a description of these kernels, I follow a standard procedure (e.g. McKenzie 1979) where I consider a small, initially spherical fluid parcel of radius $\mathbf{r}$ centred at point P located at $\mathbf{x}_p$, and subject to motion in a velocity field $\mathbf{u}$, assumed to vary slowly with time. The velocity components at a point A in the vicinity of P such that $\mathbf{x}_a = \mathbf{x}_p + \mathbf{r}$ can be approximated via a Taylor expansion truncated to first order:

$$\mathbf{u}(\mathbf{x}_a) = \mathbf{u}(\mathbf{x}_p + \mathbf{r}) \cong \mathbf{u}_p + (\nabla \mathbf{u})_p \mathbf{r}, \tag{16}$$

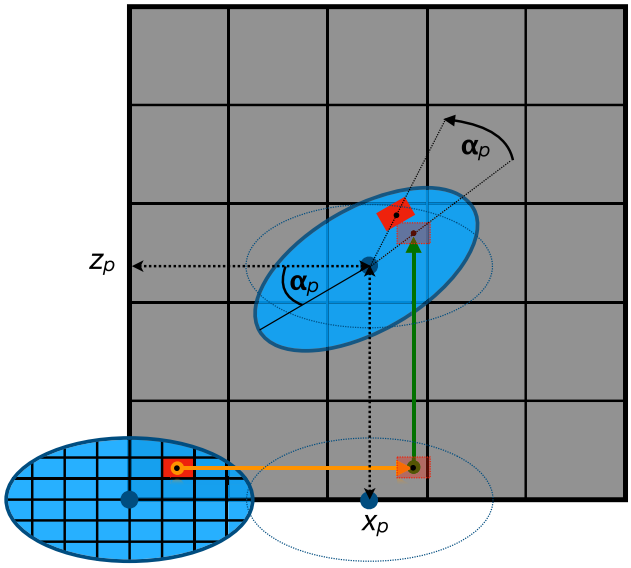

Figure 8. Schematic representation of the numerical integration procedure for the particle to mesh mapping (eq. 4). This example considers one ellipse kernel located at $\mathbf{x}_p$=(x_p, z_p) and tilted by an angle α_p with respect to the x-axis. The unrotated ellipse kernel centred at the origin $x = z = 0$ is first discretized into rectangular cells, whose local centres of mass are denoted as $\mathbf{x}_l$ (displayed in red). Then each of these centres of mass is translated by a vector $\mathbf{x}_p$ (represented by the orange and green arrows), and rotated by an angle α_p about the ellipse's centre. The volume associated with each centre of mass is added to the corresponding grid cell. Namely, a translation and rotation operator is applied to each centre of mass associated with each elementary volume of the discretized particle kernel. This procedure is applied to all particle kernels present within the domain. See text for further details.

where the subscript p refers to quantities at point P. The first term on the right-hand side of the above equation represents a translating motion that preserves the shape of fluid parcels. The second term, $\nabla\mathbf{u} = \mathbf{J}$ is the velocity gradient tensor that accounts for more complex dynamics, neglected in standard PIC approaches. It can be split into two contributions:

$$\mathbf{J} = \nabla\mathbf{u} = \mathbf{R} + \mathbf{D}, \tag{17}$$

where $\mathbf{R} = (\mathbf{J} - \mathbf{J}^T)/2$ is the antisymmetric vorticity tensor, and a symmetric deformation tensor $\mathbf{D} = (\mathbf{J} + \mathbf{J}^T)/2$. While translation and solid rotation preserve the shape of fluid parcels, shear will naturally deform the volume associated to each particle. Hence, along the fluid trajectory, $\mathbf{R}$ and $\mathbf{D}$ will rotate, stretch or shrink the vector $\mathbf{r}$ from an initial state $\mathbf{r}(t = 0) = \mathbf{r}_0$ to $\mathbf{r}(t)$. To account for such influence of $\mathbf{J}$ on the volume of the fluid parcel, one seeks a linear transformation operator, $\mathbf{M}$, that relates $\mathbf{r}(t = 0) = \mathbf{r}_0$ to $\mathbf{r}(t > 0)$:

$$\mathbf{r} = \mathbf{M}\,\mathbf{r}_0. \tag{18}$$

A relationship for $\mathbf{M}$ can be obtained by subtracting the vectors $\mathbf{r} = \mathbf{x}_a - \mathbf{x}_p$ between two instants separated by a small time increment δt: $\mathbf{r}(t+\delta t) - \mathbf{r}(t) = \mathbf{x}_a(t+\delta t) - \mathbf{x}_a(t) - [\mathbf{x}_p(t+\delta t) - \mathbf{x}_p(t)]$. Dividing this expression by δt and taking its Lagrangian limit as δt goes to 0 yields

$$\frac{D\mathbf{r}}{Dt} = \mathbf{u}_a - \mathbf{u}_p. \tag{19}$$

Combining the above relationship with eqs (16) and (18) finally yields the desired expression for the linear transformation operator $\mathbf{M}$:

$$\frac{D\mathbf{M}}{Dt} = \mathbf{J}\,\mathbf{M}, \tag{20}$$

which implies integration along fluid parcels trajectories. For a given velocity field, all the information about the evolution of a fluid parcel is contained in $\mathbf{M}$. At $t = 0$, $\mathbf{M}$ corresponds to the identity matrix $\mathbf{I}$. However, for any velocity field involving rotation and/or deformation, $\mathbf{M}$ will gradually deviate from this initial condition. Flow field involving shear will lead to the deformation of fluid parcels along preferential directions, from spheres to ellipsoids, whose semi-axis lengths correspond to the eigenvalues of $\mathbf{M}$.

The above theory can be directly applied to the PIC method, where each particle represents a fluid parcel as described by its kernel. The direct consequence is that when initiating the PIC modelling with spherically symmetric particles volumes, the particle kernels should naturally evolve towards an asymmetric form. Not accounting for such a natural evolution (i.e. assuming that $\mathbf{M} = \mathbf{I}$ for all t) unavoidably leads to sampling problems displayed in Figs 2 and 3. Instead, if the particle kernels account for deformation and rotation described by $\mathbf{M}$, these problems would be considerably reduced.

The deformable particle kernel accounts for the volume shrinking, stretching and rotation associated with each particle using a deformable ellipsoidal kernel obtained from the integration of eq. (20), with a transformation operator $\mathbf{M}_p$ associated with each particle. The contour of each particle kernel is obtained by replacing the constant r_p in eq. (11) by a function, $r_p^e = r_p\, f(\mathbf{x}_p, \mathbf{M}_p)$. For 2-D space in the (x, z) plane, r_p^e defines the contour of a tilted Lagrangian strain ellipse centred on $\mathbf{x}_p$:

$$r_p^e = r_p \sqrt{(\sigma_p^+ \cos\theta)^2 + (\sigma_p^- \sin\theta)^2 - 4\sigma_p^+\sigma_p^- \cos\theta \sin\theta \cos\alpha_p \sin\alpha_p}, \tag{21}$$

where θ ranges from 0 to π, σ_p^- and σ_p^+ are respectively the minor and major semi-axis lengths and α_p is the angle between the x-axis and the major semi-axis (i.e. the tilt of the ellipse, see Fig. 4). These quantities vary with time and for each particle because these information are derived from the transformation operator $\mathbf{M}_p$ associated with each particle. σ_p^- and σ_p^+ are the minimum and maximum eigenvalues of $\mathbf{M}_p$, and α_p is the angle between the eigenvector corresponding to σ_p^+ and the x-axis. In the case of incompressible flow, the above kernel is simplified since the volume associated to each particle is preserved, therefore $\sigma_p^- = \pi/\sigma_p^+$.

While the idea of using deformable ellipses has been proposed in Legras & Dritschel (1991), their approach was specifically designed to the modelling of vortices using a nested stack of ellipses. The method I propose here is more general, as it can be used to describe any scalar or vector field, regardless of its shape and topology. Moreover, the concept of time evolving particle kernels has also been proposed in Coppa *et al.* (1996); Bateson and Hewett (1998). However, the corresponding formalisms are less natural/physical and more complicated than the kernels proposed here, which prevents their widespread applications (Lapenta 2012).

4 IMPLEMENTATION DETAILS OF THE DPIC METHOD

In the following, I discuss practical implementation details for the application of the DPIC method. For simplicity, I restrict this discussion to the case of a domain discretized using square cells of size $h \times h$. However, the ideas developed below can be adapted to irregular grids of various dimensions and geometries.

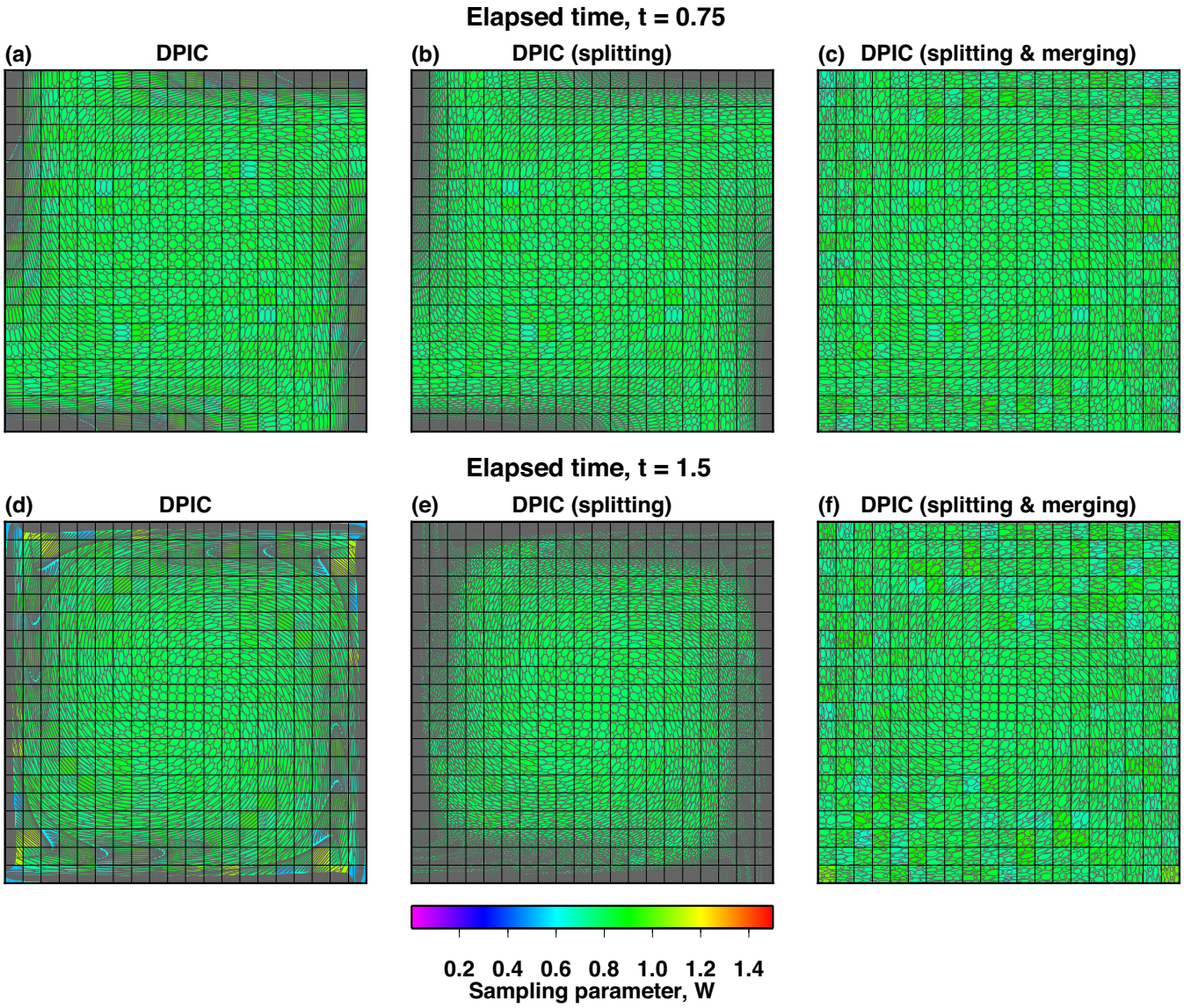

Figure 9. Results of the vortex flow test at $t = 0.75$ (top) and $t = 1.5$ (bottom), with the DPIC method using initially four particles per cell. The domain is discretized using 20×20 square cells. The cell sampling parameter W is displayed in colour. The contours of particle kernels are shown in black. Left-hand, middle and right-hand panels correspond to no particle splitting or merging, particle splitting but no particle merging is performed, particle splitting and merging is performed. See text for further details.

The general algorithm for the DPIC method consists of one initialization stage followed by five successive steps reproduced at each time step. Some steps are identical or share some similarities with the PIC method, others are specific to this new approach.

4.1 Initialization

The initiation is very similar to that of the PIC method: a set of Lagrangian particles are regularly distributed over the computational domain by assigning n_{pc0} particles per grid cell. The total number of particles (initially $n_x\, n_z\, n_{pc0}$), will evolve with time as a result of additional processes described next. As for the PIC method implemented in this work, particle kernels are initially identical and correspond to a disc of radius $r_p = h/4 = r_p^e$. However, in order to minimize particle kernel overlaps no random perturbation is applied to $\mathbf{x}_p$ (Fig. 5a). Therefore, each particle kernel is tangent to its four closest neighbours (Fig. 5a). Note that different, possibly more compact, arrangements could be considered. For example, the alternate horizontal (or vertical) shifting of particles rows/columns of identical disc kernels by a distance $r_p/2$ would produce an hexagonal packing (Fig. 5b). This would yield a more compact distribution of the disc kernels, and would increase the number of tangent points between particle disc kernels to six instead of four. In addition, any initial particle kernel arrangement could be supplemented by sets of smaller particles to fill the gaps without overlaps, leading to values of W even closer to the ideal value of one (Figs 5c–d). Preliminary tests seem to indicate that for the same total number of particles, this does not yield significant improvements compared to simpler arrangements (Fig. 5a). One could also consider possible particle overlaps in order to increase W. However, our tests suggest that the resulting solution accuracy tends to be reduced. Nevertheless, these alternatives require more extensive and systematic investigations in the future, in order to find optimum initial arrangements. Contrary to the PIC method where only one scalar parameter (generally a

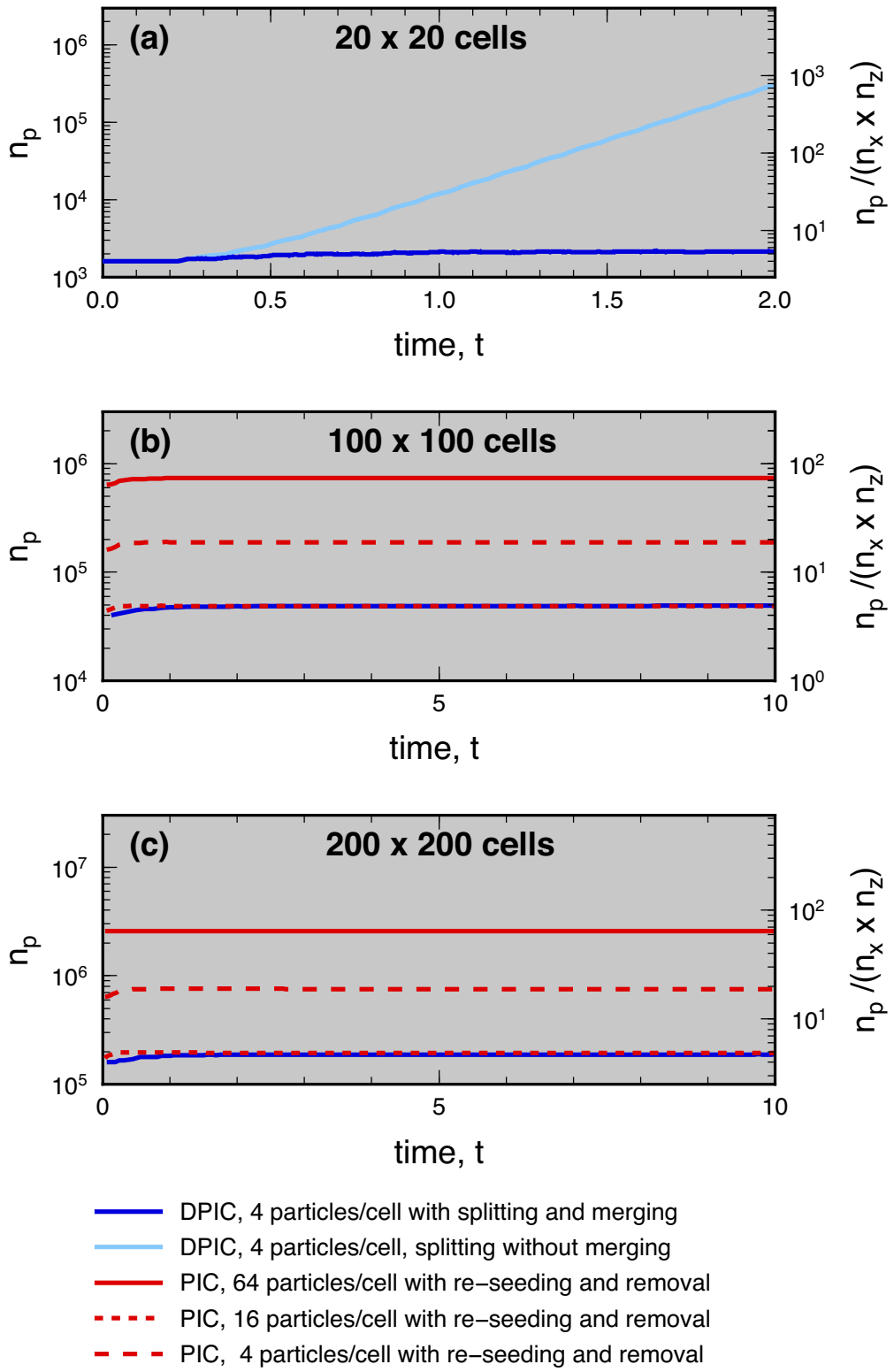

Figure 10. Results of the vortex flow test. Time evolution of the number of particles n_p (or, n_{pc}, the value of average number of particles per cell) for different grid sizes and different methods.

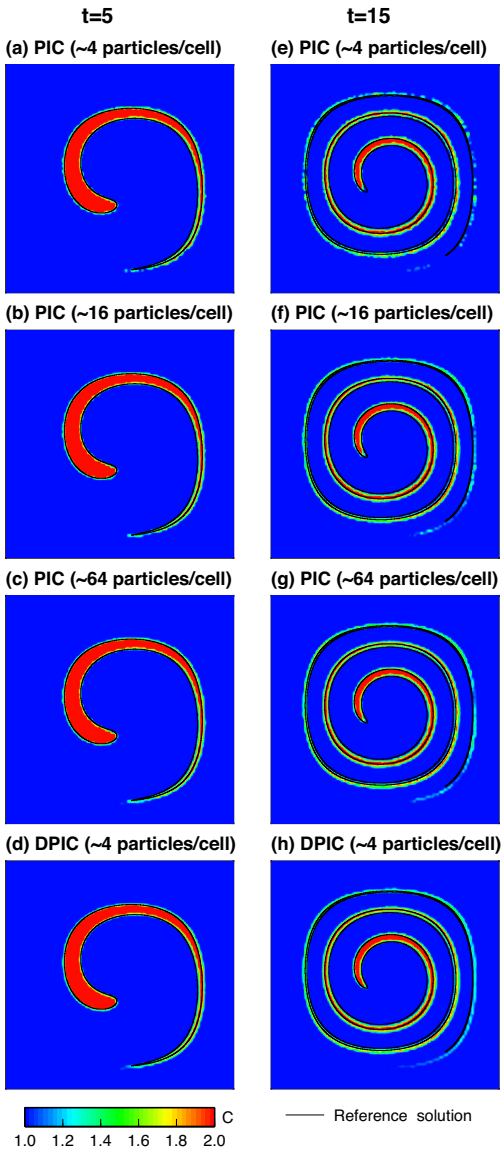

Figure 11. Results of the vortex flow test: values of the scalar field on the Eulerian grid, in a domain discretized using 100 × 100 square cells at time $t = 5$ (left) and $t = 15$ (right). The top three panels correspond to the PIC method with randomly perturbed initial positions of particles and particle remeshing, and using initially 4, 16 and 64 particles per cell. The bottom panel corresponds to the DPIC method with initially four particles per cell.

radius, r_p) is associated to kernels, each particle is also associated with a 2×2 matrix (a 3×3 matrix in 3-D space) $\mathbf{M}_p$. As mentioned in Section 3, $\mathbf{M}_p$ is initialized to the identity matrix.

4.2 Particle advection

As in the standard PIC method, particles' positions are advanced in time using a TVD-RK scheme, with velocities at particles' locations determined via conservative velocity interpolation (Meyer & Jenny 2004; Wang *et al.* 2015). The time steps satisfy the same CFL-based time step considered for the PIC method. This procedure is therefore strictly identical to the one applied for the PIC implemented in this study.

4.3 Particle kernel computation

The entries of $\mathbf{M}_p$ are updated by solving eq. (20). The entries of the velocity gradient tensor $\mathbf{J}$ are computed with the same interpolation scheme for velocities used during the particle advection step.

From the new values of $\mathbf{M}_p$, one can determine σ_p^- and σ_p^+. In practice, these are obtained by taking the square roots of the eigenvalues of the symmetric matrix: $\mathbf{M}_p^{\mathrm{T}}\mathbf{M}_p$, the right Cauchy–Green strain tensor (Farnetani & Samuel 2003). The angle α_p between σ_p^+ and the x-axis is also derived from $\mathbf{M}_p$ following the approach detailed in Fuchs & Schmeling (2013).

4.4 Particle splitting

The formalism on which the DPIC method is based (see Section 3) assumes that the dimensions of the fluid parcels represented by the particle kernels are small. However, if one lets evolve deformable particle kernels in a flow involving shear, the kernel's semi-major axis will continuously grow, possibly at an exponential rate, and will quickly reach values that are significantly larger than the dimensions of the grid cells, or even the size of the physical domain Ω. To prevent this, a splitting of the largest particle kernels into smaller kernels may be performed. This leads to a variation in the total number of particles with time. The splitting procedure is displayed in Fig. 6. After experimentation with different criteria, the procedure is applied to a given particle if the latter meets the following requirements based on the aspect ratio and the maximum size of its corresponding kernel:

$$\sigma_p^+/\sigma_p^- > 5 \; or \; \sigma_p^+ > 2. \tag{22}$$

When the above criterion is met, the corresponding 'parent' particle characterized by its location $\mathbf{x}_{p_0}$ and its corresponding $\mathbf{M}_{p_0}$ is removed and replaced by two new 'son' particles. The parent kernel is split into two identical ellipses corresponding to two new particles, which will only differ by their locations $\mathbf{x}_{p_1}$ and $\mathbf{x}_{p_2}$. The son particles have the following positions:

$$\mathbf{x}_{p_1} = \mathbf{x}_{p_0} + \frac{r_p \sigma_{p_0}^+}{2}(\cos\alpha_{p_0}, \sin\alpha_{p_0})^{\mathrm{T}} \tag{23a}$$

$$\mathbf{x}_{p_2} = \mathbf{x}_{p_0} - \frac{r_p \sigma_{p_0}^+}{2}(\cos\alpha_{p_0}, \sin\alpha_{p_0})^{\mathrm{T}}, \tag{23b}$$

and are associated with identical kernel values:

$$\sigma_{p_1}^+ = \sigma_{p_2}^+ = \frac{1}{2}\sigma_{p_0}^+ \tag{24a}$$

$$\sigma_{p_1}^- = \sigma_{p_2}^- = \sigma_{p_0}^- \tag{24b}$$

$$\alpha_{p_1} = \alpha_{p_2} = \alpha_{p_0}. \tag{24c}$$

If, upon applying eqs (24a)–(24c), $\sigma_{p_{1,2}}^+ < \sigma_{p_{1,2}}^-$ the values of the semi-axis lengths for the son particles are swapped with each other and $\alpha_{p_{1,2}} = \alpha_{p_{1,2}} + \pi/2$.

This simple splitting procedure therefore results in two identical son ellipsoids with area twice smaller than that of the parent kernel, and identical tilt (except for the case described just above). This allows the conservation of the area represented by the parent and son kernels. Eq. (23) implies that they will be tangent to each other at the location $\mathbf{x} = \mathbf{x}_{p_0}$. Note that this splitting procedure may result in kernel overlaps between the son particles and the rest of the particle population. However, tests have confirmed that overlaps remain statistically limited. Nevertheless, particles for which kernel

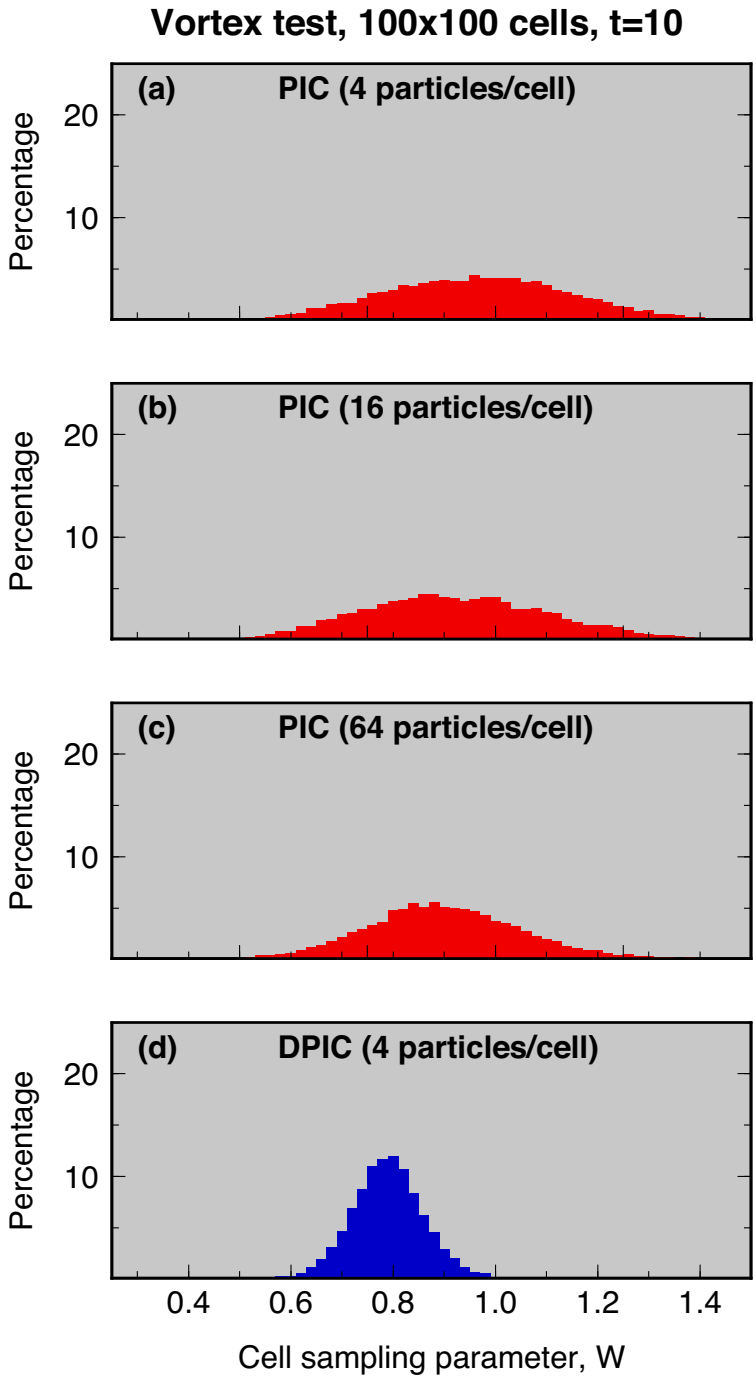

Figure 12. Results of the vortex flow test at $t = 10$, in a domain discretized using 100×100 square cells. Histograms of the cell sampling parameter, W, corresponding to different initial values of particle number per cell, n_{pc0}. (a–c) PIC method with random perturbed initial positions of particles and particle reseeding and removal with n_{pc0}= 4, 16 and 64, respectively. (d) DPIC method with particle splitting and merging with $n_{pc0} = 4$.

overlaps become important will be merged through a procedure described in the next section.

With the knowledge of $\sigma_{p_1}^+ = \sigma_{p_2}^+ = a, \sigma_{p_1}^- = \sigma_{p_2}^- = b$, and $\alpha_{p_1} = \alpha_{p_2} = \theta$ for a pair of son particles, their corresponding orthogonal eigenvectors are $\mathbf{v}_p^+ = a\,(\cos\theta, \sin\theta)^{\mathrm{T}}$ and $\mathbf{v}_p^- = b(-\sin\theta, \cos\theta)^{\mathrm{T}}$. Then, assuming incompressibility, the matrix entries for the pair of son particles are deducted from the fact that $(\mathbf{M}_p - \sigma_p^+ \mathbf{I})\mathbf{v}_p^+ = 0$ and that $(\mathbf{M}_p - \sigma_p^- \mathbf{I})\,\mathbf{v}_p^- = 0$. This leads to a system of four equations whose solution is

$$\mathbf{M}_{p_1} = \mathbf{M}_{p_2} = \mathbf{M}_p = \begin{bmatrix} a\cos^2\theta + b\sin^2\theta & (a-b)\sin\theta\cos\theta \\ (a-b)\sin\theta\cos\theta & b\cos^2\theta + a\sin^2\theta \end{bmatrix}. \tag{25}$$

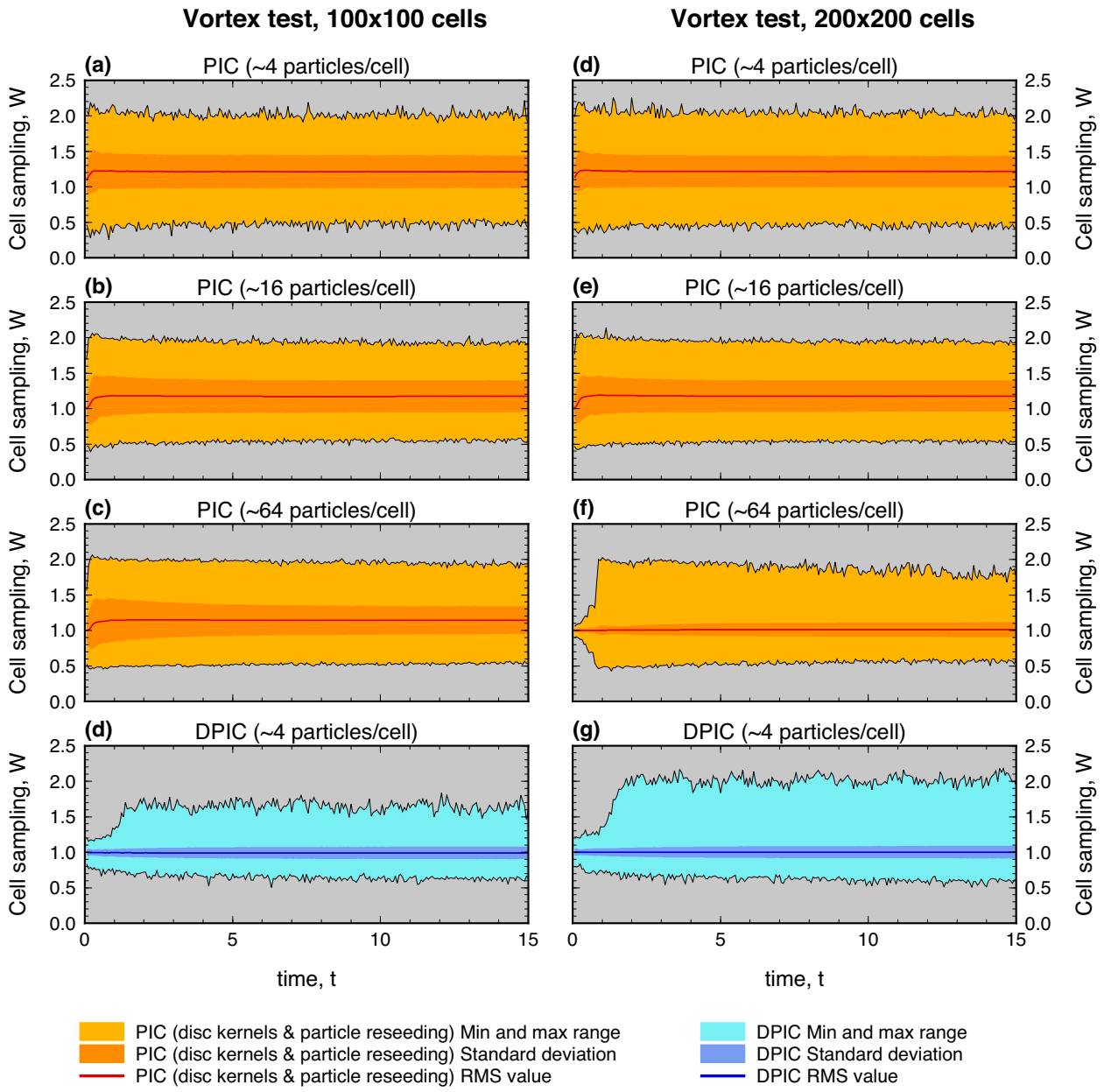

Figure 13. Results of the vortex flow test, in a domain discretized using 100 × 100 (left) and 200 × 200 (right) square cells. Time evolution of the cell sampling parameter, W corresponding to different initial values of particle per cell, n_{pc0}. The top three panels correspond to the PIC method with randomly perturbed initial positions of particles and particle remeshing with n_{pc0}= 4, 16 and 64 particles per cell. The bottom panel corresponds to the DPIC method with n_{pc0}=4. The RMS value is represented by the thick curve, the min and max values are displayed by thin black lines and the standard deviation corresponds to the dark coloured area.

4.5 Particle merging

The splitting procedure described above naturally yields an increase with time of the number of particles, which can rapidly degenerate into a prohibitive associated computational cost. In addition, this could result in an increase of kernel overlaps between the new particles and the remaining population, which may yield an over-representation, thus a source of inaccuracy, together with a waste of computational resources. These problems can be avoided by applying a merging procedure in regions where the particle concentration is too important, as schematically shown in Fig. 7. The procedure uses a finer grid, whose corresponding spacing in each direction is r_p, superimposed on the Eulerian grid. Within each cell of this finer grid, we count the number of particles. Since the grid considered here is regular this can be performed in a cost-effective way via a single sweep through the particles' positions. If more than two particles per fine grid cell are found, the following merging procedure is applied to the particles whose positions belong to the cell. Particles that meet this criterion for a given fine grid cell of centroid $\mathbf{x}_f$ are denoted as $E(\mathbf{x}_f)$. For these particles, a local fine-grid cell weight, w_l, is calculated. The latter corresponds to the ratio of the area represented by the particle kernel divided by the total area of the particle kernels that belong

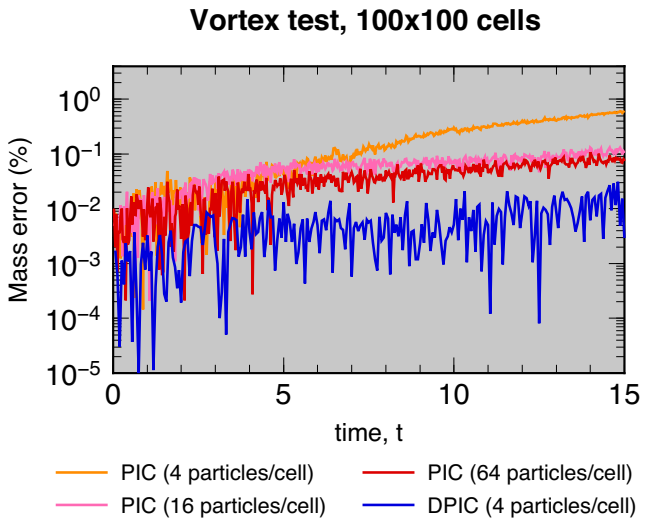

Figure 14. Results of the vortex flow test. Time evolution of the mass error for a domain discretized using 100 × 100 square cells.

to $E(\mathbf{x}_f)$:

$$w_l = \frac{\sigma_p^+ \sigma_p^-}{\sum_{p|\mathbf{x}_p \in E(\mathbf{x}_f)} \sigma_p^+ \sigma_p^-}. \tag{26}$$

The corresponding set of particles will subsequently be deleted and replaced by a single particle, with a kernel of larger size, whose centre is determined via arithmetic averaging:

$$\mathbf{x}_p(\mathbf{x}_f) = \sum_{p|\mathbf{x}_p \in E(\mathbf{x}_f)} \mathbf{x}_p w_l. \tag{27}$$

Among the various possibilities to determine the merged ellipse dimensions, the simplest and most efficient approach was found to require that the aspect ratio of the merged ellipse is the arithmetic mean over the corresponding set of merged particle kernels:

$$\frac{\sigma_p^+(\mathbf{x}_f)}{\sigma_p^-(\mathbf{x}_f)} = \sum_{p|\mathbf{x}_p \in E(\mathbf{x}_f)} \frac{\sigma_p^+}{\sigma_p^-} w_l. \tag{28}$$

Assuming incompressibility, the above requirement simplifies to the following expression for the semi-major axis of the merged ellipse:

$$\sigma_p^+(\mathbf{x}_f) = \sqrt{\sum_{p|\mathbf{x}_p \in E(\mathbf{x}_f)} (\sigma_p^+)^2}. \tag{29}$$

In order to preserve the total area delimited by the deleted particle kernels, the value of the semi-minor axis for the new merged kernel is set as

$$\sigma_p^-(\mathbf{x}_f) = \frac{\sum_{p|\mathbf{x}_p \in E(\mathbf{x}_f)} \sigma_p^+ \sigma_p^-}{\sigma_p^+(\mathbf{x}_f)}. \tag{30}$$

The tilt of the merged ellipse, $\alpha_p(\mathbf{x}_f)$, is obtained via weighted arithmetic averaging:

$$\alpha_p(\mathbf{x}_f) = \sum_{p|\mathbf{x}_p \in E(\mathbf{x}_f)} \alpha_p w_l. \tag{31}$$

The transformation operator for the new merged particles is then evaluated using eq. (25) with $a = \sigma_p^+(\mathbf{x}_f)$, $b = \sigma_p^-(\mathbf{x}_f)$ and $\theta = \alpha_p(\mathbf{x}_f)$. Again, more accurate merging procedures could be implemented, for example, by checking explicitly whether particle kernels overlap prior to merging, but this would significantly increase the computational cost. Although there is probably room for further improvement, the proposed merging procedure was found to efficiently maintain approximately constant the total number of particles in the presence of splitting, while allowing for a good kernel sampling of the computational domain, as we shall see in the next sections.

The approach described above was implemented via two consecutive sweeps through the population of particles. In the frame of pure advection considered in this paper (eq. 1), and for the non-trivial case where the field **C** to advect has more than one distinct value, several populations of particles are considered (one per distinct value). The merging procedure described above is applied separately to particles of different types. If the field to advect was more smoothly varying, as in the case of the advective and diffusive transport using a Fluid-Implicit-Particle (FLIP) approach (Brackbill *et al.* 1987; Gerya & Yuen 2003), merging could occur without restrictions in particle type, but this case is beyond the purpose of this study.

Note that in order to maintain a balance with the splitting procedure, merging is not allowed to occur more than twice per fine grid cell and per time step, with however one exception. To avoid excessive accumulation of small particle kernels within some cells, a supplementary merging procedure is only applied to cells characterized by a bulk sampling parameter W larger than 1.5. During this specific merging procedure, all the kernels sharing the same identity and whose centre belong to the excessively oversampled cell are merged all at once. Since this procedure is followed by splitting, it does not necessarily bound W below 1.5 but it was found to be sufficiently efficient to prevent any excessive kernel clustering.

Finally, while both merging or splitting alter $\mathbf{M}_p$ and $\mathbf{x}_p$, the constant value of r_p initially set for each particle remains unaffected by these procedures. This does not mean that the kernel sizes remain unchanged since σ_p^- and σ_p^+ are affected by the merging and splitting procedures as explained above.

Note that the use of particle merging and splitting based on Lagrangian strain has also been proposed in the frame of the PIC method in order to reduce non-homogeneous sampling of the domain by particles (Moresi *et al.* 2003). However, these procedures were only associated with point-wise particle kernels, and used a simplified approach to estimate the Lagrangian strain.

4.6 Conversion from Lagrangian to Eulerian grid values

The implementation of eq. (4) can be performed via two end-member approaches. A first possibility is to evaluate the particles' weights w_* in eq. (6) exactly by analytical integration of the particles kernels' areas within a given grid cell bounded by four corners (see Appendix B for further details).

Despite its machine precision accuracy, this approach requires the extensive use of conditional statements and calls for costly square root and trigonometric functions, leading to an important computational extra cost. In addition, extension of this approach to 3-D space would be much more complicated. For these reasons, an alternative approach was considered for evaluating the particles' weights w_* via a simple numerical integration of the area span by a particle elliptical kernel in a given cell. In this case, the ellipse is rotated by an angle $-\alpha_p$, and discretized into smaller rectangular cells. The area corresponding to each of these smaller cells is added to the corresponding grid cell, based on the location of the centre of

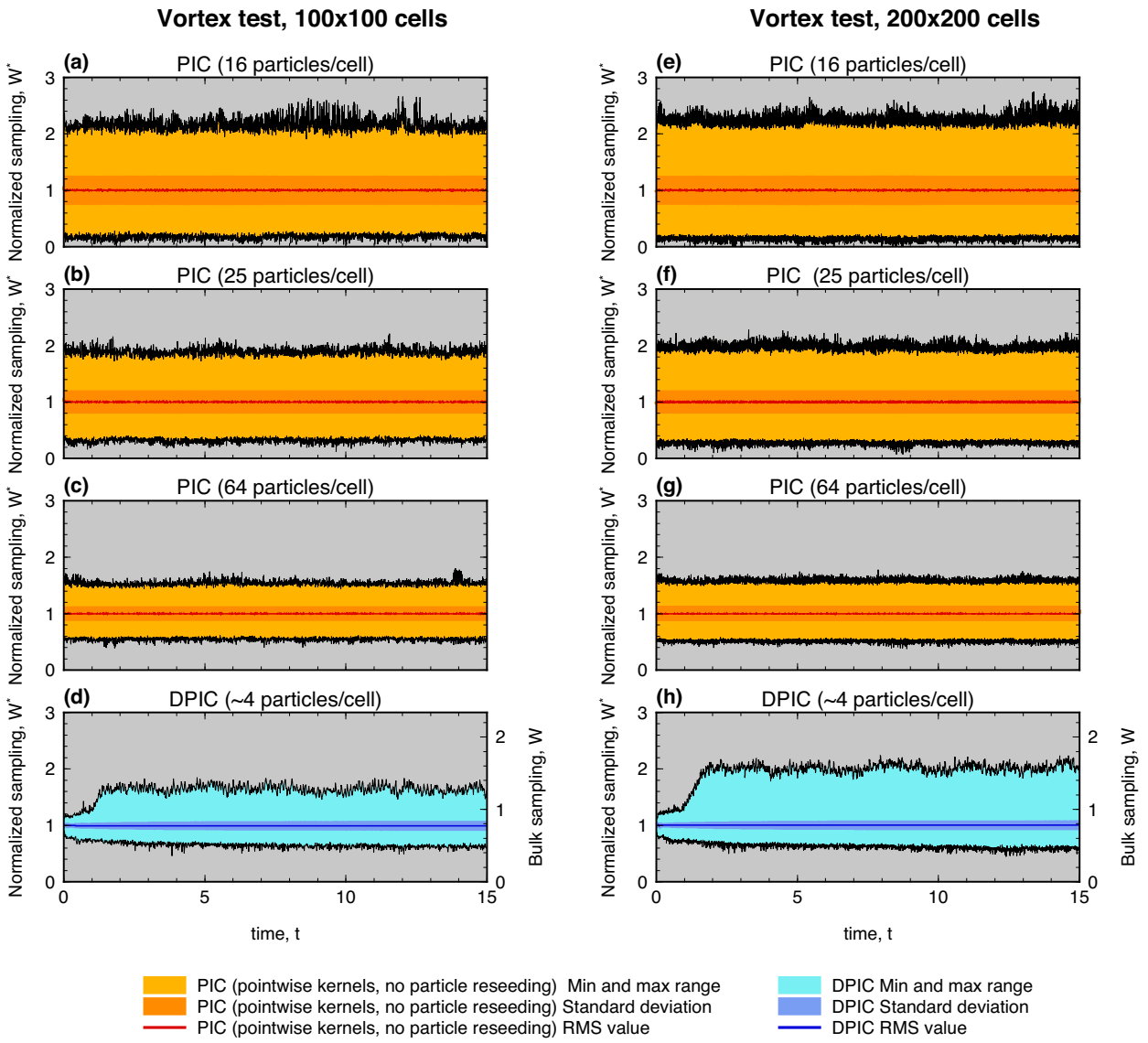

Figure 15. Results of the vortex flow test, in a domain discretized using 100 × 100 (left) and 200 × 200 (right) square cells. Time evolution of the normalized cell sampling parameter, W^* corresponding to different initial values of particle numbers. The three top panels correspond to the PIC method using point-wise kernels, no particle remeshing, with 16, 25 and 64 particles per cell. The bottom panel corresponds to the DPIC method with $n_{pc0} = 4$. The RMS value is represented by the thick curve, the min and max values are displayed by thin black lines, and the standard deviation corresponds to the dark coloured area.

mass of these smaller cells within the grid (see Fig. 8). Using three or four cells per semi-axis length was found to be a good compromise between accuracy and computational efficiency, with results very similar to those obtained using the analytic approach. Although the elliptical deformable kernels are anisotropic, their remaining degree of symmetry can be exploited to reduce the computational cost associated with the numerical integration of the area overlap between particle kernels and grid cells. Only a complete swipe through one quarter of the elliptical kernel surface (bounded by the ellipse minor and major semi-axis) is required. Then, many of the operations can be duplicated at a lower cost in the remaining three quarters.

Hybrid numerical–analytical approaches are also possible but have not been tested.

To significantly improve the computational efficiency, it is desirable to use additional computational arrays that store once per RK step the values of $\cos\alpha_p$ and $\sin\alpha_p$ for each particle required for a number of calculations (e.g. eq. 25) rather than calling for intrinsic trigonometric functions on the fly.

The methods described above remain valid for the case of disc particle kernels, and are therefore directly applicable for classic PIC approaches, with however several simplifications (e.g. no kernel rotation is required in this case). However, for PIC cases using point-wise kernels, particle–mesh mapping was performed using the same expressions as in Gerya and Yuen (2003), which are more efficient due to a considerably smaller computational complexity.

5 SPATIAL SAMPLING FOR THE DPIC METHOD

I reproduced the experiment considered in Section 2.4 using the DPIC method. As for the PIC method, four particles per grid cell

are initially regularly distributed in the 2-D domain discretized using 20×20 square cells. Three implementations were considered: (1) splitting and merging procedures are not applied, (2) splitting is applied but no merging procedure is applied and (3) the full implementation of the DPIC method: both particle kernel splitting and merging are applied.

Fig. 9 displays the results at the same early and later stages shown in Fig. 2. In the absence of particle splitting and merging (Figs 9a and d), the total number of particles remains constant with time. However, due to the deformation imposed by the velocity field, the particles that travel through the regions of most intense deformation (i.e. the four corners of the domain) experience a strong kernel stretching with time, while the rest of the particles' population located more inwards remains more isotropic (although no longer circular). Contrary to the PIC method, the deformable particle kernels remain tangent to each other without much overlap during the early stage. This results in a homogeneous spatial sampling of the domain by the particles. However, due to their intense deformation, particle kernel overlap eventually develops because the σ_p^+ values become too large, which violates the assumption of small fluid parcel dimensions stated in Section 3. This shows that particle kernel splitting is required in order to maintain accurate results. When splitting alone is accounted for, particle kernel overlaps are removed at all times (Figs 9c and e). However, accounting for splitting without merging results in an exponential increase of the number of particles with time in regions of intense deformation (displayed in Fig. 10a). This leads to a drastic increase of the computational cost, which depends primarily on the total number of particles. To maintain the computational cost at an acceptable level, the splitting can be supplemented by the merging procedure previously described. In this case, we observe some kernels overlaps. However, the latter remain moderate and considerably smaller than what was obtained with the best results of the PIC method (compare Figs 2c and e with Figs 9c and e). For a similar number of particles, the DPIC method yields a significantly more homogeneous spatial sampling of the domain than the PIC method, with much smaller variations in values of the sampling parameter. The splitting combined with the merging procedure maintains the number of particles approximately constant with time, as illustrated in Fig. 10(a).

This feature was observed for all the tests described in this paper, as well as in other experiments not shown, even with larger amounts of deformation on larger portions of the model domain, and for various grid resolutions. In general, for flows characterized with larger amounts of deformation, more important variations in the number of particles per cell are observed, but the spatial sampling of the domain by the particle kernels remains homogeneous.

6 COMPARISON OF THE PIC AND DPIC METHOD FOR 2-D STEADY AND UNSTEADY FLOWS

The tests performed in the previous section clearly demonstrate the superiority of the DPIC method over the PIC method for spatial sampling. However, additional experiments are required to confirm that such an improvement in spatial sampling translates into an improved accuracy for the DPIC method. For this reason, I consider in this section four additional tests that consist in following the evolution of a scalar field with two distinct values in steady kinematic, time-dependent kinematic or fully dynamic flows.

6.1 Steady vortex flow test

For this test, the same velocity field (eq. 14) used in Section 2.4 is considered, in a square domain [0,1]×[0,1] discretized with 100 × 100, or 200 × 200 square cells. Initially, the scalar field has a value $C = 2$ within a circular disc of radius 0.15, whose centre is located at ($x = 0.5$, $z = 0.75$), and $C = 1$ elsewhere. Fig. 11 displays the results obtained at an early ($t = 5$) and a later ($t = 15$) stage, using 100 × 100 cells with the PIC method with disc kernels (eq.11, with $r_p = h/4$), and initially $n_{pc0} = 4$, 16 and 64 particles per cell, or using the DPIC method with initially four particles per cell. A reference solution obtained using a high-precision marker chain front tracking algorithm (Samuel & Bercovici 2006) is also shown. Since both particle reseeding and removal (for the PIC method) and splitting and merging (for the DPIC method) are applied, the total number of particles remains roughly constant with time for each case considered. This can be seen in Figs 10(b) and (c) for two grid resolutions. Results obtained with the PIC method using initially four particles per cell strongly suffer from non-homogeneous sampling with 'spotty' features clearly visible, even at the early stage. With a fourfold increased number of particles the sampling problems are reduced but remain clearly visible. A further increase in the number of particles to an average of 64 per cell yields results that do not appear strongly affected by sampling problems. These results compare well with those obtained using the DPIC method, with initially only four particles per cell (4.7 on average for these cases). This underlines the benefit of using deformable particle kernels. Fig. 12 shows histograms of the sampling parameter at $t = 10$ for the same cases displayed in Fig. 11. It confirms that if the number of particles is too small, the PIC method suffers from large variations in spatial sampling by the particles, as seen with the large width of the histograms for 4 and 16 particles per cell. When 64 particles per cell are used, the sampling for the PIC method is closer (but remains of lower quality) than sampling obtained using the DPIC method with much fewer particles (Figs 12c and d). This comparison holds for all times and for different values of the grid resolution, as shown in Fig. 13. The latter displays the time evolution of the sampling parameter: Root Mean Squared (RMS) value, and its variability (standard deviation, min and max values) corresponding to the cases depicted in Figs 11 and 12, for a domain discretized using either 100 × 100 or 200 × 200 square cells. For all cases shown in Fig. 13, standard deviation, min and max values for W quickly reach a statistically constant values. However, the standard deviation values for W are significantly larger for the PIC method using too few particles (4 or 16 per cell initially), which reflects the development of spatial sampling problems. In addition, both the maximum amplitude variation, and standard deviation for W at all times and for both grid resolutions, decrease with increasing the number of particles. As seen previously, all cell sampling statistics shown in Fig. 13 compare well for the PIC method using 64 particles per cell and the DPIC method using only four particles per cell.

Fig. 11 suggests that the accuracy of the DPIC method is better than that of the PIC method using the same amount or even a greater number of particles.

To measure the amount of numerical dissipation introduced within each method, I computed the mass error at time t for each case:

$$e_{\text{mass}} = \frac{\left| \int_\Omega C(\mathbf{x}, t)\mathrm{d}\Omega - \int_\Omega C(\mathbf{x}, t_0)\mathrm{d}\Omega \right|}{\int_\Omega C(\mathbf{x}, t_0)\mathrm{d}\Omega}. \tag{32}$$

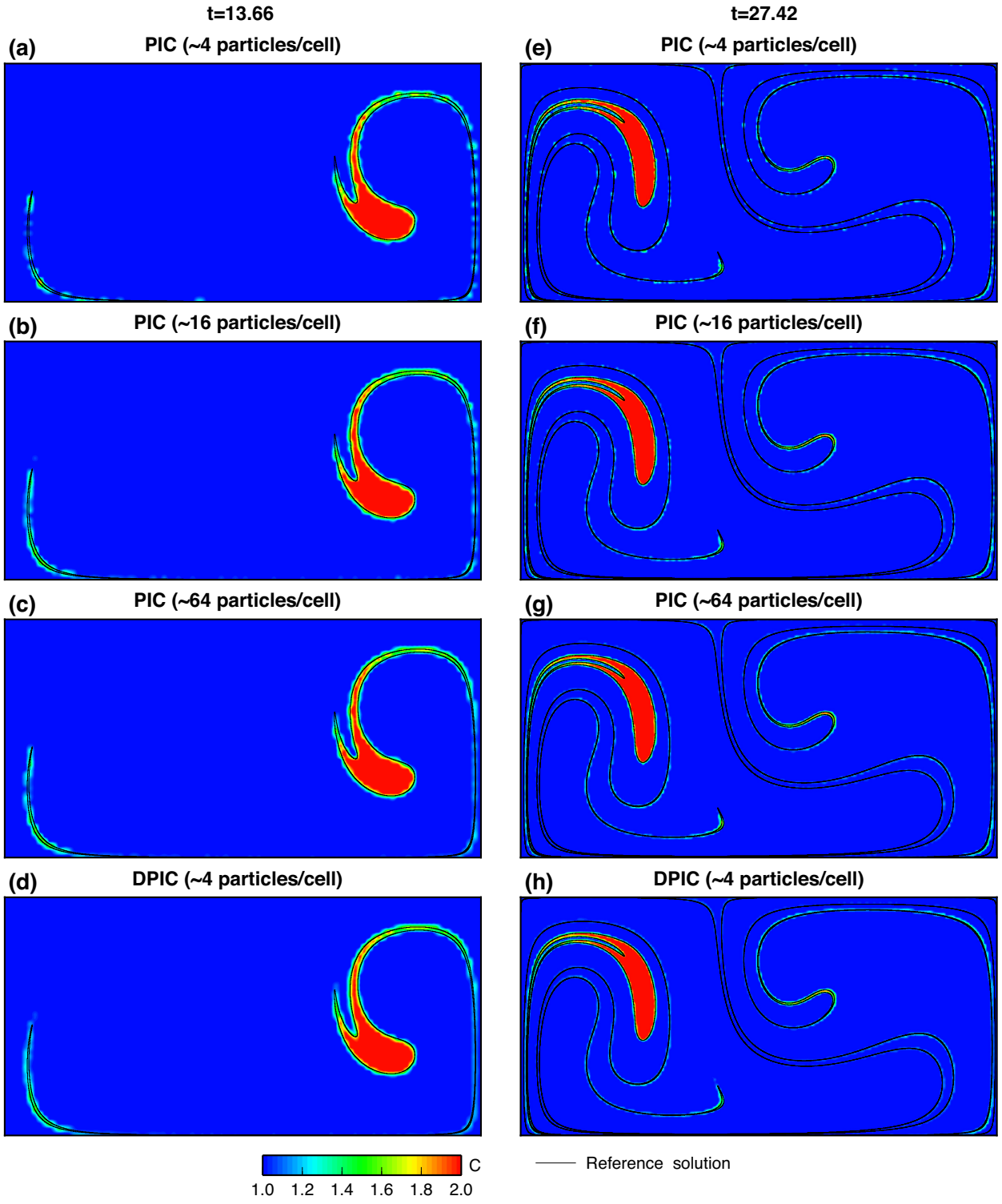

Figure 16. Results of the chaotic flow test: values of the scalar field on the Eulerian grid, composed of 200 × 100 square cells at time $t = 13.66$ (left) and $t = 27.42$ (right). The three top panels correspond to the PIC method with randomly perturbed initial positions of particles and particle remeshing with initially 4, 16 and 64 particles per cell. The bottom panel corresponds to the DPIC method with initially 4 particles per cell.

Fig. 14 displays the time evolution of the mass error for the cases considered in Fig. 13 corresponding to domains discretized using 100 × 100 square cells. It confirms that the improved spatial sampling of the DPIC method results in a smaller error than the PIC solution using the same (or even a larger) amount of particles, as also seen in Fig. 11. The mass error for the PIC method using initially 4 or 16 particles per cell shows a significant and continuous increase with time, up to values around 0.1 per cent (with 16 particles per cell), and about 0.5 per cent (for four particles per cell) at final time $t = 15$. On the contrary, both the value of e_{mass} and its increase with

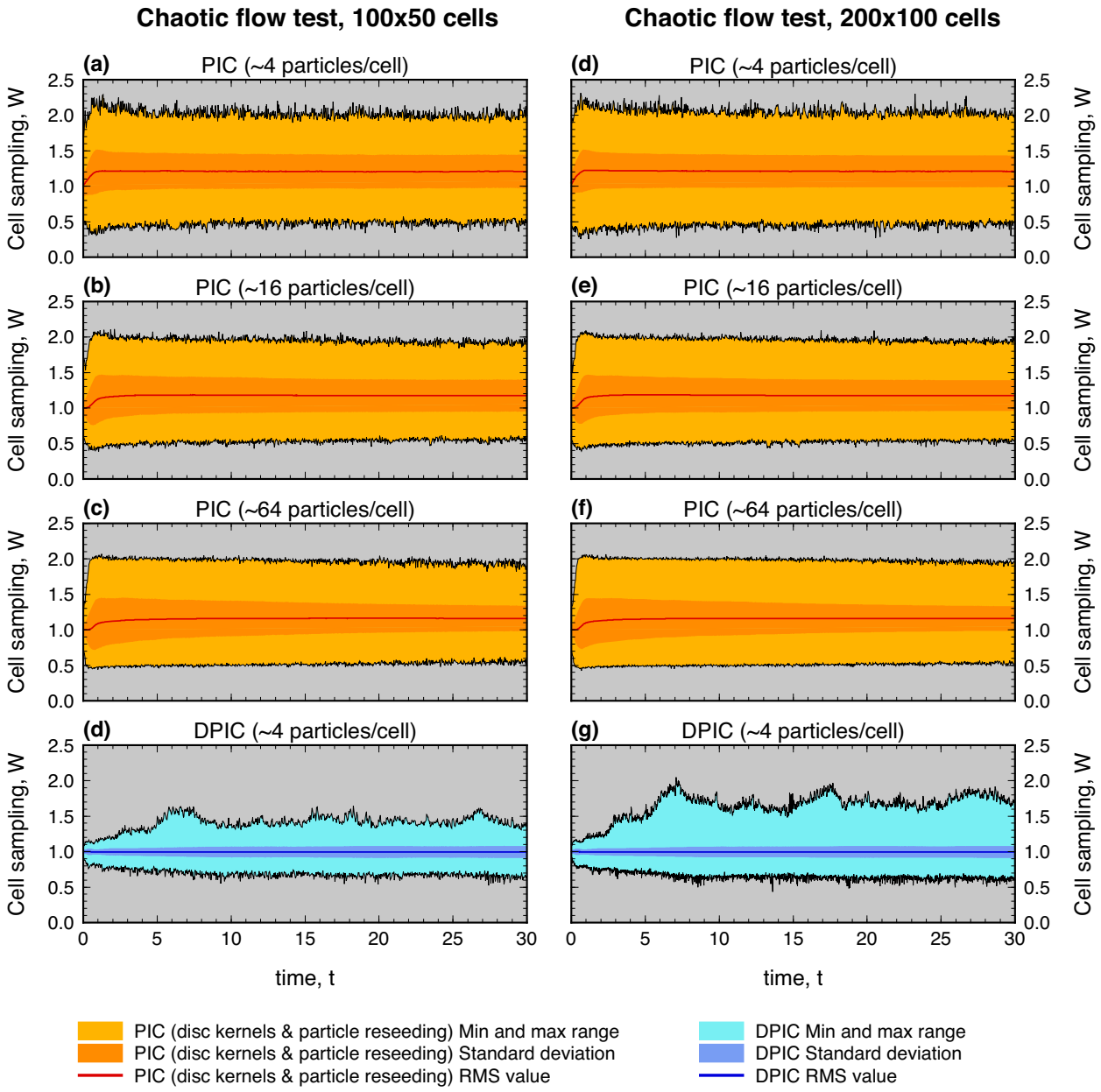

Figure 17. Results of the chaotic flow test, in a domain discretized using 100 × 50 (left) and 200 × 100 (right) square cells. Time evolution of the cell sampling parameter, W corresponding to different initial values of particle numbers n_{p0}. The three top panels correspond to the PIC method with random perturbed initial positions of particles and particle remeshing with initially 4, 16 and 64 particles per cell. The bottom panel corresponds to the DPIC method with initially four particles per cell. The RMS value is represented by the thick curve, the min and max values are displayed by thin black lines and the standard deviation corresponds to the dark coloured area.

time are much smaller for the DPIC method using four particles per cell. In order to reach comparable accuracy, the PIC method requires more than 64 particles per cell.

A set of PIC experiments with point-wise kernels commonly used in geodynamic modelling was also performed. In these experiments, no particle reseeding or removal is applied. In the case where empty cells develop, one linearly interpolates from the closest particles in order to obtain a value of the compositional field in the empty cell. As systematically done in this study, conservative velocity interpolation is used upon particle advection. Cases using 16, 25 and 64 particles per cell were considered.

Fig. 15 displays the time evolution of the normalized cell sampling, W^*, for all cases, including the DPIC already shown in Fig. 13(d). Cases using 16 particles per cell result in the development of empty cells ($W^* = 0$), on both coarse and finer grids (Figs 15a and e). Similar to what is observed in Fig. 13, the PIC and DPIC results become comparable when the number of particles is 16 times greater than that of the DPIC method. In fact, PIC cases using 64 particles per cell still yield a less homogeneous sampling (smaller min values, and larger standard deviation of the normalized cell sampling values, W^*) than the DPIC method using four particles per cell (compare Figs 15c and d & Figs 15g and h). It seems that the PIC method using disc kernels and particle reseeding yields slightly better results, at the cost, however, of considerably heavier computations.

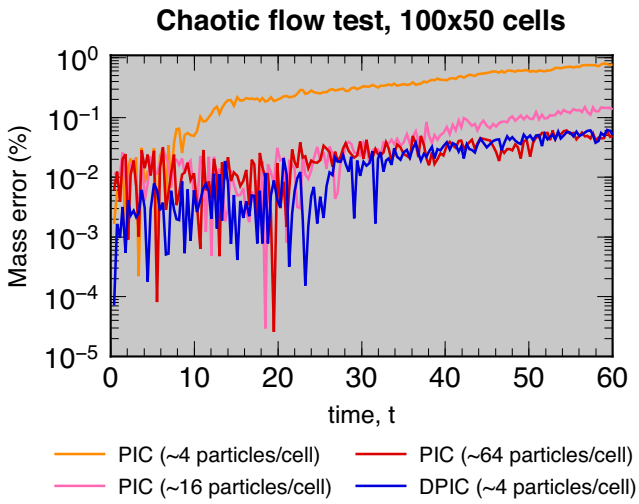

Figure 18. Results of the chaotic flow test. Time evolution of the mass error for a domain discretized using 100 × 50 square cells.

6.2 Time-dependent chaotic flow test

To extend the previous test to time-dependent velocity fields I considered the following double-gyre chaotic flow field in a 2-D domain [0,2]×[0,1] (Brunton & Rowley 2010):

$$\mathbf{u} = \left(-\frac{\pi}{10} \sin(\pi f_g) \cos(\pi z),\ \frac{\pi}{10} \cos(\pi f_g) \sin(\pi z) \frac{df_g}{dx} \right)^T, \quad (33)$$

where f_g is a function of the x coordinate and a periodic function of time:

$$f_g(x,t) = \frac{1}{4} \sin\left(\frac{2\pi}{10} t\right) x^2 + x - \frac{1}{2} \sin\left(\frac{2\pi}{10} t\right) x. \quad (34)$$

Within the domain, we track the evolution of a scalar field C whose initial value is set to 2 inside a disc of radius 0.15 and of centre ($x = 0.2, z = 0.5$), and 1 elsewhere. Fig. 16 displays the results of the experiment at two elapsed times for a domain discretized using 200 × 100 square cells. The solution obtained using the DPIC method with initially four particles per cell, and the results obtained using the PIC method with 4, 16 and 64 particles per cell are shown. A reference solution calculated using a very accurate front tracking marker chain algorithm delineates the interface between the two scalar values of C. The prescribed chaotic flow reduces the size of the disc into long thin filaments stretching at an exponential rate. As for the vortex flow test (Section 6.1), experiments performed using the PIC method with too few particles (Figs 16a and e) yield visibly inaccurate 'spotty' results. While solutions obtained using the PIC method with 16 particles per cell instead of four are visibly more accurate (Figs 16a and b & Figs 16e and f), the overall accuracy remains poor compared to that of the DPIC method using only four particles per cell, as illustrated by the time evolution of the mass error for this experiment (Fig. 18). The improved accuracy of the DPIC method results again from a more homogeneous spatial sampling of the domain by the particles, as seen in Fig. 17. As also observed for the steady vortex flow test, this figure shows the same improvement in spatial sampling for the DPIC compared to that for the PIC method. Here, also 64 particles per cell are necessary for the PIC method to improve the sampling, and to reduce the error to the same level than that of the DPIC method with four particles per cell (Fig. 18).

As in the previous test, I performed a set of PIC experiments with point-wise kernels and no particle reseeding or removal, and either 16, 25 or 64 particles per cell. In addition, I considered a case using 16 particles per cell for which velocity interpolation to advect the particles was replaced by the use of the exact analytical expression for the velocity field. This allows the impact of velocity interpolation on the spatial sampling of the PIC method to be evaluated. Fig. 19 displays the time evolution of the normalized cell sampling, W^*, for all cases, including the DPIC already shown in Fig. 17(d). Similar to what was observed for the vortex test, cases using 16 particles per cell result in the development of empty cells ($W^* = 0$), on both coarse and finer grids (Figs 19a and b & Figs 19f and g). No notable differences are found between the cases using the exact expression of velocity for particles advection, and those using conservative velocity interpolation (compare Figs 19a and b & Figs 19f and g). The persistence of sampling problems for cases that are not prone to significant inaccuracies in particles advection confirms that non-homogeneous sampling originates from a different cause than the quality of particles advection only. In fact, it is due to the fact that point-wise kernels cannot account for deformation in the vicinity of each particle contrary to the DPIC method. Similar to what is observed in Fig. 17, the PIC and DPIC results become comparable when the number of particles is 16 times greater than that of the DPIC method. However, here again PIC cases using 64 particles per cell yield a less homogeneous sampling than that of the DPIC method using four particles per cell (compare Figs 19d and e & Figs 19i and j).

Overall, the vortex and chaotic kinematic experiments detailed above show that the DPIC method yields a far better spatial sampling than the PIC method, leading to a significant improvement in solution accuracy (2–3 orders of magnitude mass error reduction) compared with the PIC method using comparable or even larger amounts of particles, regardless of the grid resolution. The DPIC method with only four particles per cell yields results of comparable accuracy than the PIC method using 64 particles per cell.

6.3 SolCx flow test

I have considered the benchmark test based on the so-called SolCx analytical solution of the Stokes flow in presence of strong viscosity contrasts (Zhong 1996) and used in a number of studies (Duretz *et al.* 2011; Thielmann *et al.* 2014; Wang *et al.* 2015; Pusok *et al.* 2016). The setup is identical to that in Wang *et al.* (2015) and Pusok *et al.* (2016). The square unit domain is decomposed into 32 × 32 identical square cells, and the SolCx solution [as implemented in the 'Underworld' package (Moresi *et al.* (2007)] is imposed on the nodal gridpoints. The left half of the domain is characterized with a viscosity of 1, while the remaining half has a viscosity of 10^4. Fig. 20 shows the normalized sampling after 5000 time steps obtained with the PIC method using point-wise kernels and different amounts of particles. As seen in Wang *et al.* (2015), the use of the conservative interpolation scheme of Meyer and Jenny (2004) prevents strong particle clustering or rarefaction. In addition, the values of W^* are visibly more homogeneous when the number of particles increases. The results obtained with the DPIC method using four particles per cell on average are also displayed and show a generally more homogeneous sampling than the PIC method using less than 64 particles per cell on average. However, one can note the presence of a few cells near the viscosity jump with both relatively large or small values of W^*. These were also observed for the same settings in experiments using the PIC method and conservative velocity interpolation schemes (Pusok *et al.* 2016), depending on the interpolation scheme used. In our case, the oversampled cells

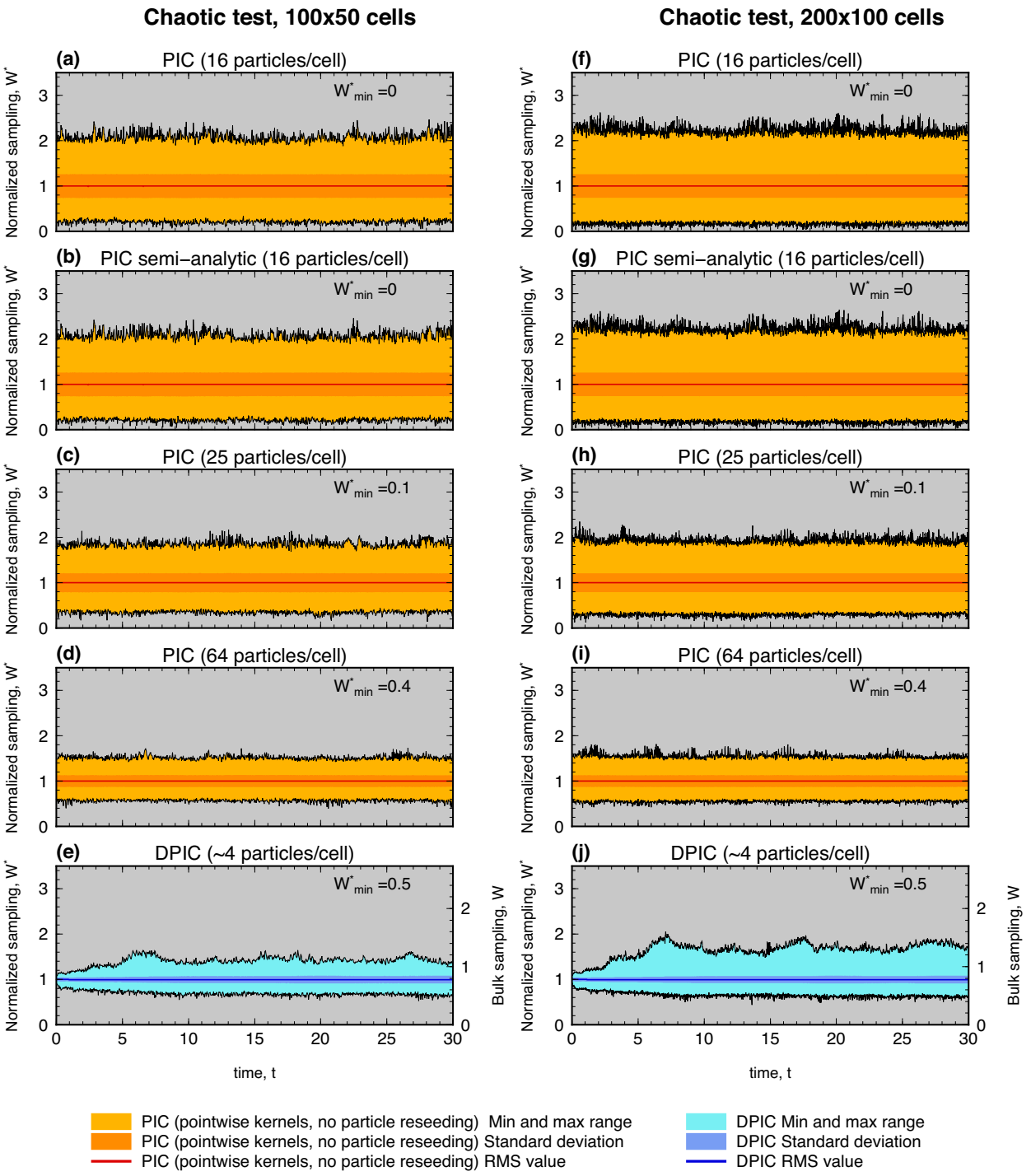

Figure 19. Results of the chaotic flow test, in a domain discretized using 100 × 50 (left) and 200 × 100 (right) square cells. Time evolution of the normalized cell sampling parameter, W^* corresponding to different initial values of particle numbers. The three top panels correspond to the PIC method using point-wise kernels, no particle remeshing, with 16, 25 and 64 particles per cell. The bottom panel corresponds to the DPIC method with initially four particles per cell. The RMS value is represented by the thick curve, the min and max values are displayed by thin black lines and the standard deviation corresponds to the dark coloured area.

mostly result from the splitting procedure, which tends to generate an artificial displacement of particles in some areas. Indeed, when a particle kernel is split into two smaller particles (see Fig. 6), the newly created kernels have their centres of mass distinct from that of the parent kernel. This shift in centre of mass can be seen as an artificial instantaneous displacement of the particles. Possible way to fix/minimize this would be to consider a different type of splitting, generating for instance three new particles instead of two: a central, larger one surrounded by two smaller kernels. The centre of mass of the largest kernel would remain identical to that of the parent kernel, thereby limiting the observed shift upon splitting. Nevertheless, the clustering induced in this experiment remains bounded, thanks to

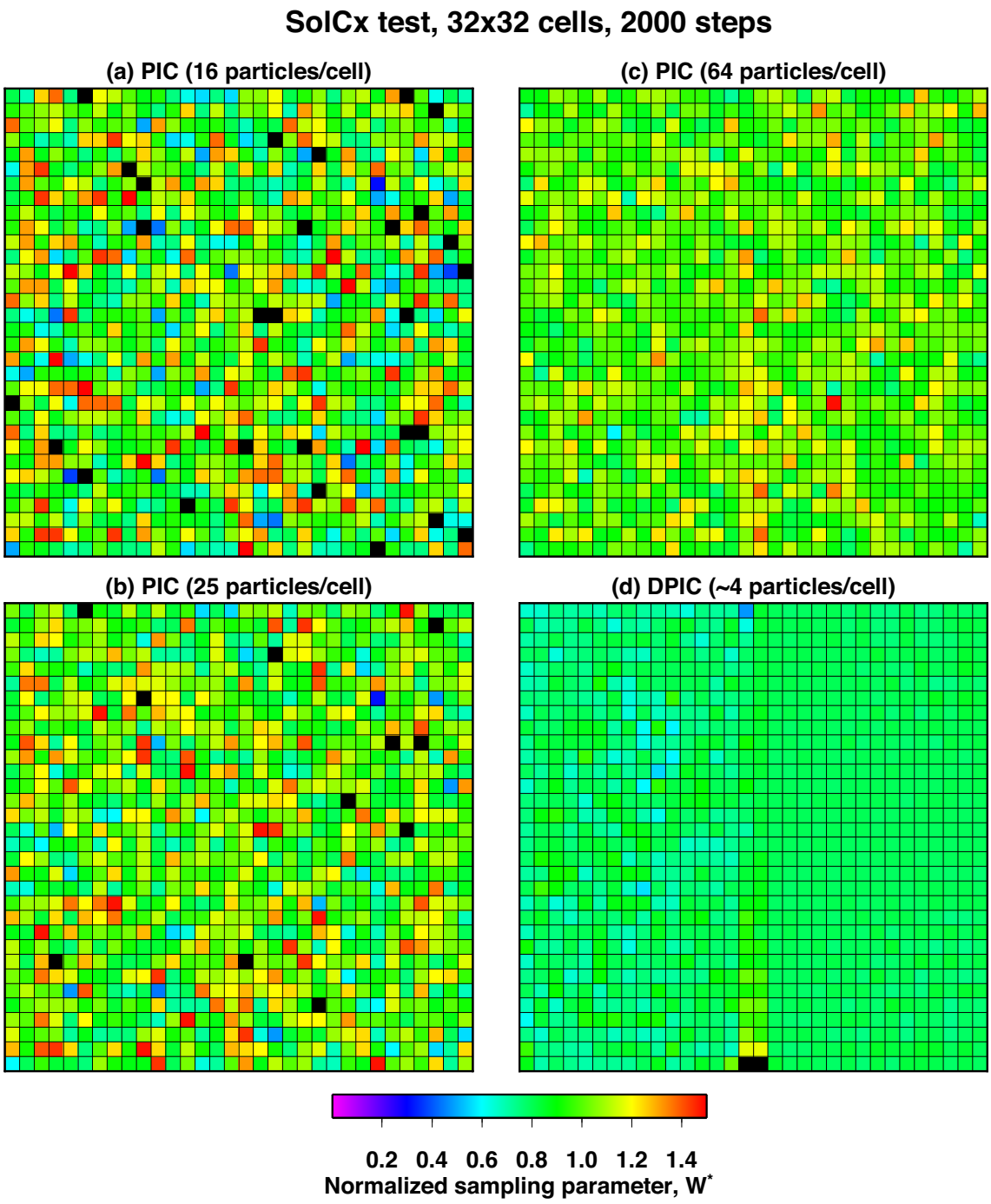

Figure 20. Results of the SolCx test after 2000 time steps. Normalized cell sampling obtained for three PIC cases using point-wise kernels and different amounts of particles (a–c). Results obtained using the DPIC method with about four particles per cell. Black cells represents $W^* > 1.4$.

the merging procedure. This can be seen in Fig. 21 where the peak in sampling values is periodically removed upon application of the merging procedure in oversampled areas. Even for much longer time periods, maximum values for bulk cell sampling for the DPIC method remain essentially below 2.0. The other metrics displayed in Fig. 21 also confirm that the DPIC method with only four particles per cells generates a sampling of comparable quality than the PIC method with 16 times more particles (e.g. compare the standard deviation for W^* for both cases).

6.4 Dripping instability test

The previous experiments and benchmarks presented above are all based on kinematic flows where the velocity field at grid locations is known exactly, and the compositional field associated with the particles, is purely passive (i.e. it does not influence the flow). To complete the set of tests, I have therefore considered a purely dynamic experiment where the velocity is computed as the solution of the Stokes equations represented by the following set of dimensionless equations for the conservation of mass and momentum,

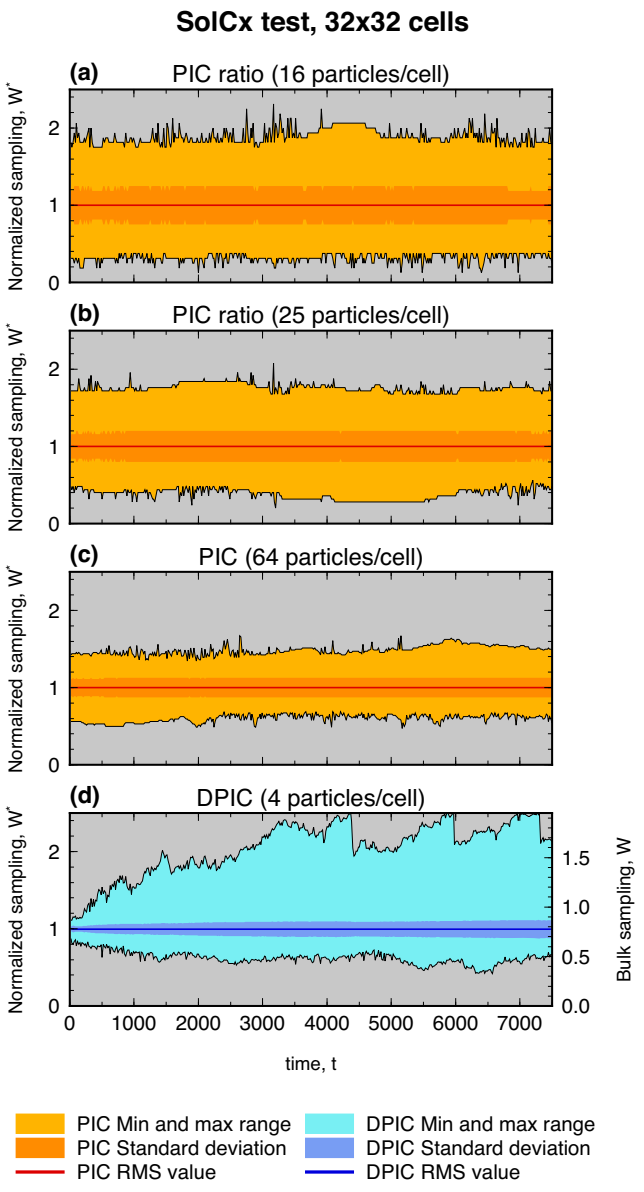

Figure 21. Results of the SolCx test. Time evolution of the normalized cell sampling parameter, W^* corresponding to different initial values of particle numbers. The three top panels correspond to the PIC method using pointwise kernels, no particle remeshing, with initially 16, 25 and 64 particles per cell. The bottom panel corresponds to the DPIC method with initially four particles per cell. The RMS value is represented by the thick curve, the min and max values are displayed by thin black lines, and the standard deviation corresponds to the dark coloured area.

respectively:

$$\nabla \cdot \mathbf{u} = 0, \tag{35}$$

$$-\nabla p + \nabla \cdot \eta(\nabla \mathbf{u} + \nabla^T \mathbf{u}) + Rb\, C \mathbf{e}_z = 0. \tag{36}$$

In the above equations, p is the dynamic pressure, η the dynamic viscosity, $\mathbf{e}_z$ is a vertical unit vector pointing upward and Rb is compositional Rayleigh number set to one. These equations are actively coupled to eq. (1) via the buoyancy term $Rb\, C \mathbf{e}_z$ present in the conservation of momentum.

Several dynamic benchmarks for PIC methods in geodynamics have been proposed in past studies (van Keken *et al.* 1997; Gerya & Yuen 2003; Wang *et al.* 2015; Pusok *et al.* 2016). However, they require either complex rheologies and/or setting (e.g. Wang *et al.* 2015; Pusok *et al.* 2016) or they were found to be insufficiently demanding to reveal differences between PIC and DPIC methods. Instead, I considered a simple setting that was found to be sufficiently demanding for both PIC and DPIC methods. The experiment is sketched in Fig. 22 and consists in the Rayleigh–Taylor destabilization of a dense material of rectangular shape, located at the top of the domain. Viscosity is set to one in the upper half of the domain and increases abruptly to 200 in the lower half. Horizontal surfaces are rigid, while free-slip boundary conditions are applied on vertical side walls. The [0,1] × [0,2] domain was discretized using either 50×100 or 100×200 square cells, in which eqs (35) and (36) where recast using a pure streamfunction formulation and solved using the finite-volume code 'StreamV' (Samuel & Evonuk 2010; Samuel 2012b) benchmarked against various analytical and numerical solutions (Samuel 2012a; Tosi *et al.* 2015). Calculations were performed until reaching the dimensionless time $t = 1.4\,10^5$, corresponding roughly to 1600 time steps, depending on the cases. This setup is particularly demanding because the rigid horizontal boundaries combined with a rheological boundary at mid-depth triggers the presence of stagnation points and the associated pure shear, which tend to generate sampling problems in PIC methods, as discussed in Section 2.4. I performed experiments using the PIC

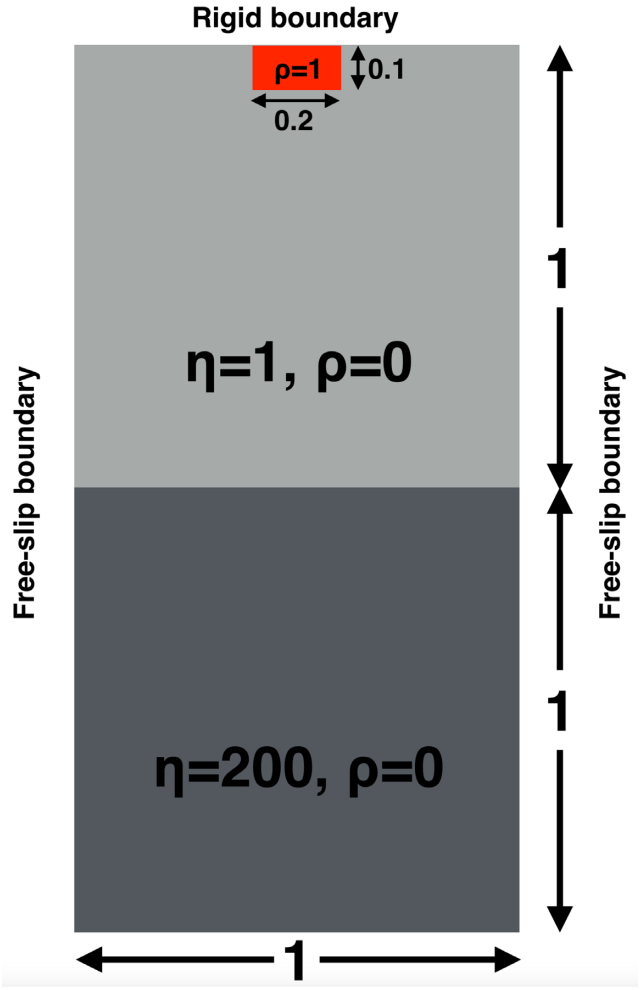

Figure 22. Schematic representation of the dripping instability test. See text for further details.

method with 16, 25 and 64 particles per cell, and point-wise kernels, and two experiments using the DPIC method with either four particles per cell or about three particles per cell with a more compact arrangement (Fig. 5b). Fig. 23 shows three snapshots in time obtained on a 100×200 grid using the PIC method with 16 particles per cell and the DPIC method with four particles per cell. The less homogeneous spatial sampling of the PIC method yields a less continuous/more 'spotty' compositional field with a pronounced asymmetry towards the end of the evolution (Fig. 23c). This asymmetry results from the combination of random initial position of particles, amplified by the development of under and oversampled areas. Indeed, one can see in Fig. 23(d) the presence of an empty cell located at the boundary between the dense and the regular material, where most of the deformation takes place. On the contrary, the DPIC method with four times less particles yields a much more continuous and symmetric compositional field (Figs 23e–g). The deformable kernels allow a better spatial sampling, as can be seen in Fig. 23(h), which represents the same region shown in Fig. 23(d), at a comparable sinking distance of the dense material. One can note that even the cell characterized by the smallest sampling hosts the centre of mass of 10 particles, not mentioning the contribution of kernels whose centre of mass are located elsewhere. This multiplication of particles in regions characterized with strong deformation results from the splitting procedure. This considerably reduces the chances of the development of empty cells. Fig. 24 shows the time-evolution of the spatial sampling for all cases. On the coarse grid, the PIC method with less than 64 particles per cell yields the development of both empty and strongly oversampled cells. The DPIC method performs similarly or better than the best results of the PIC method, but using 16–20 times less particles. On the finer grid, the sampling problems of the PIC method are reduced (e.g. no empty cells develop for the case using about 25 particles per cell). However, here again, 16–20 times more particles are necessary to yield a spatial sampling comparable to that of the DPIC method (Figs 24f–i).

This fully dynamic test illustrates the superiority of the DPIC method over the PIC method, as observed in the kinematic tests discussed earlier. Despite its simplicity, the setup considered here can be relevant to a number of geodynamic scenarios such as, subduction, delamination of an eclogitic root, differentiation within a magma chamber or core formation in terrestrial bodies.

Overall, for all the tests presented in this study, the DPIC method with an average of two to five particles per cell always yields a spatial sampling greater than 0 at all times. However, this does not necessarily guarantee that in some extreme situations, involving the combination of low grid resolution, small number of particles and strong localized deformation, the development of empty cells can be avoided with this method. However, when this happens the occurrence of empty cell remains considerably more limited than that of the PIC method using a greater amount of particles (64 or more).

7 COMPUTATIONAL COST

While the PIC and the DPIC methods share common operations (particles advection, particle–mesh mappings), they also use different procedures, which can lead to a distinct computational cost. In this section I compare the performances and the distribution of the computational load for the PIC and the DPIC methods.

For a given number of particles, the additional procedures of the DPIC method (splitting, merging, kernel update and elliptic kernels-to-grid mappings) yield a computational extra cost increasing the execution time by a factor 3–4 relative to PIC method.

The computational cost for both methods is almost directly proportional to the total number of particles. This quasi-linear dependence is shown in Fig. 25 that displays the elapsed time corresponding to the scalar execution of the dripping instability test using 100 × 200 grid cells for 1400 time steps. Note that the differences in timing performances between the PIC and the DPIC methods would remain comparable if I had considered other tests. Despite the fact that some procedures (e.g. kernel merging or spitting) are applied only at the end of each RK cycle, no significant differences were found between second- or third-order RK time integration.

The distribution of the computational load among the main procedures for the DPIC and the PIC methods are displayed in Tables 1 and 2, respectively. While the kernels-to-grid mapping represents the most time-consuming procedure of the DPIC algorithm, it may be more difficult to further optimize it significantly. However, kernel merging, splitting and update, which altogether represent 42 per cent of the time spent can certainly be further optimized, for instance, by using more efficient particle sorting techniques and ordering, such as linked cell approaches (Welling & Germano 2011). Note that contrary to the DPIC method, the particle-to-grid mapping in the PIC method is the least consuming part of the algorithm. This is due to the fact that the use of point-wise kernels prevents from the area integration of the particle kernels.

Despite the extra cost associated with the use of deformable kernels relative to the use of point-wise kernels, the improved sampling in the DPIC method with only three to four particles per cell yields results that are comparable or better than the PIC method with 64 particles per cell. Therefore, at comparable accuracy, the DPIC method is four to six times more efficient than the PIC method using point-wise kernels, commonly used for geodynamic modelling. In addition, the tests conducted in the previous sections all indicate that a minimum of 25 particles per cell for the PIC method is required to avoid spurious sampling problems such as the development of empty cells. As seen in Fig. 25, the DPIC method already becomes more efficient than the PIC method when the number of particles per cell becomes larger than 16. Therefore, even with this minimum requirement, the use of the DPIC method is preferable, given the fact that particle operations often represent the most consuming part of geodynamic calculations (Tackley 2008).

8 CONCLUSIONS

I have presented a new evolution of the PIC method based on the use of elliptical deformable kernels that account for the Lagrangian strain history in the vicinity of the particles. Such deformable kernels are directly related to the original idea proposed in Harlow (1957) as macroscopic fractions of fluid/material surrounding each particle, which is physically sound. These deforming kernels allow for a much more homogeneous spatial sampling of the domain by particles, compared to standard 1-D fixed-shape kernels. The latter leads to spatial over- and undersampling that degrade the accuracy of the numerical solution with time, or require a prohibitive amount of particles in order to get acceptable results. The use of deformable particle kernels comes with an extra computational cost. However, such extra cost is acceptable, considering the gain in accuracy of the DPIC method compared to that of the PIC method, regardless of the type of kernel (disc or point-wise) used. The DPIC method

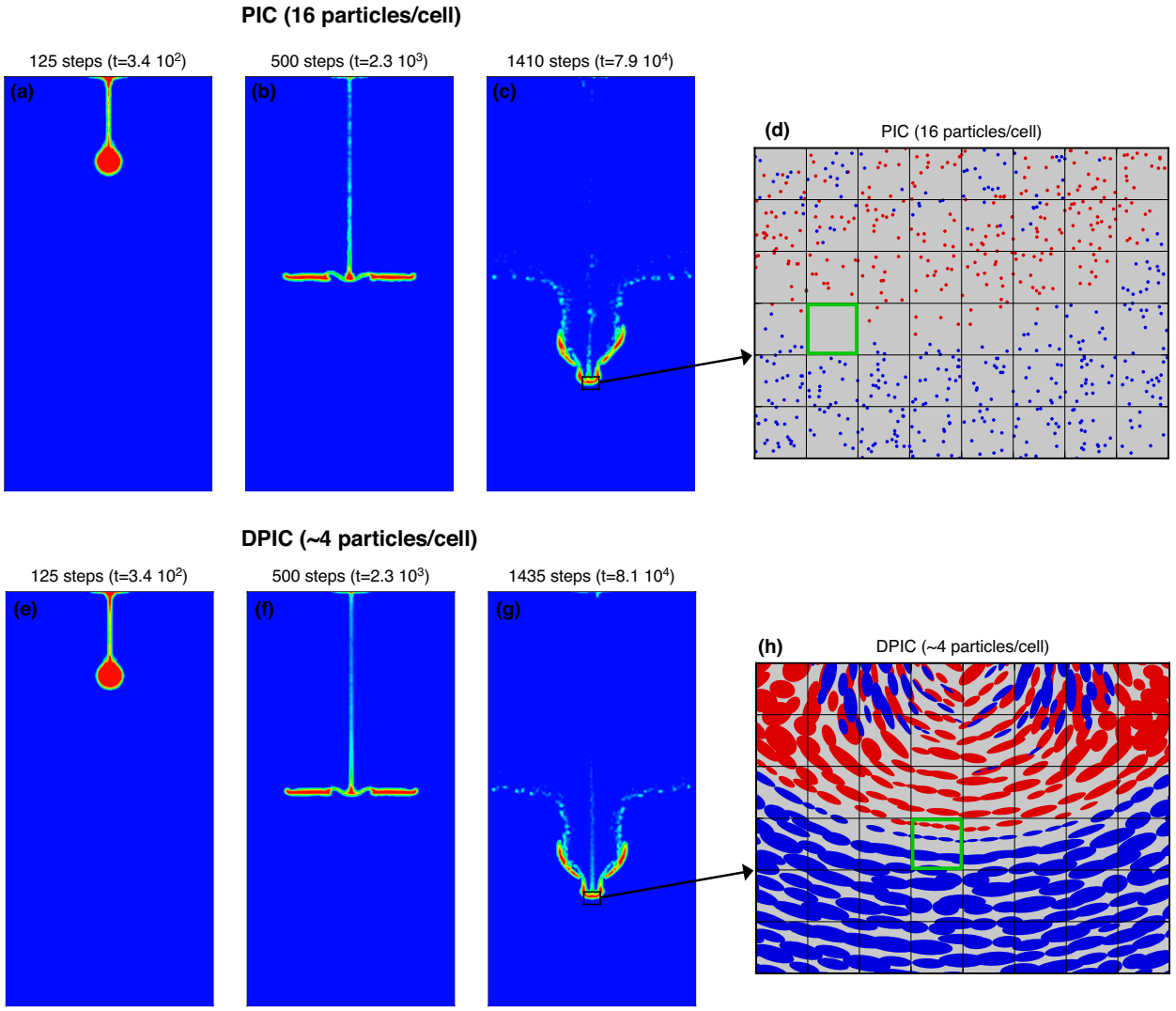

Figure 23. Results of the dripping instability test on a domain discretized using 100×200 cells. Snapshots in time of the compositional field obtained with the PIC method using 16 particles per cell (a–c) or with the DPIC method using four particles per cell (e–g). Closed up views of the area delimited by the black rectangles in panels (c) and (g) are shown in panels (d) and (h), respectively. These show the particle kernels. The green squares indicate the location of the cell the least sampled by particle kernels.

yields a dynamically evolving number of particles through the use of merging and splitting procedures. These procedures allow maintaining the number of particles approximately constant. In addition, despite the changes in particle number, merging and splitting procedures do not involve interpolation of the quantities carried by the particles, contrary to particles remeshing used in PIC methods. This contributes to the smaller numerical dissipation of the DPIC method.

While the DPIC method presented here focused on the case of pure advection of discontinuous quantities in 2-D domains, this new approach could be generalized to 3-D space where the computational gain could be even more substantial. Indeed, 2-D tests have shown that the accuracy of the DPIC method using two particles per dimension is comparable with that of the PIC method using four times more particles per dimension. If we extrapolate this rule-of-thumb to 3-D space, comparable accuracy could be obtained between the DPIC method using eight particles per cell instead of 512 particles per cell for the PIC method. On the other hand, extensions of the operations associated with the DPIC method from 2-D to 3-D space would likely lead to an increase in computational cost (in particular for operations associated with kernel update and kernel-to-grid mappings) possibly by a factor 2–4 per particle. Overall, at comparable precision and taking everything into consideration, the estimated computational savings in using the DPIC method in 3-D geometry relative to the PIC method would be comparable or greater to that observed in 2-D. In addition, the DPIC method could be adapted to cases where non-advective transport is present, by combining the approach used in the FLIP method (Brackbill *et al.* 1987) with the use of elliptical/ellipsoidal kernels. Finally, the DPIC method could be implemented in the frame of variable grid spacing, a common situation in geodynamic modelling, by using smaller particle kernels in finer portions of the grid (e.g. Fig. 5), and accounting for variable grid spacings during the splitting and merging procedures. This would allow the number of particles per cell to be comparable to that for the case of constant grid spacing.

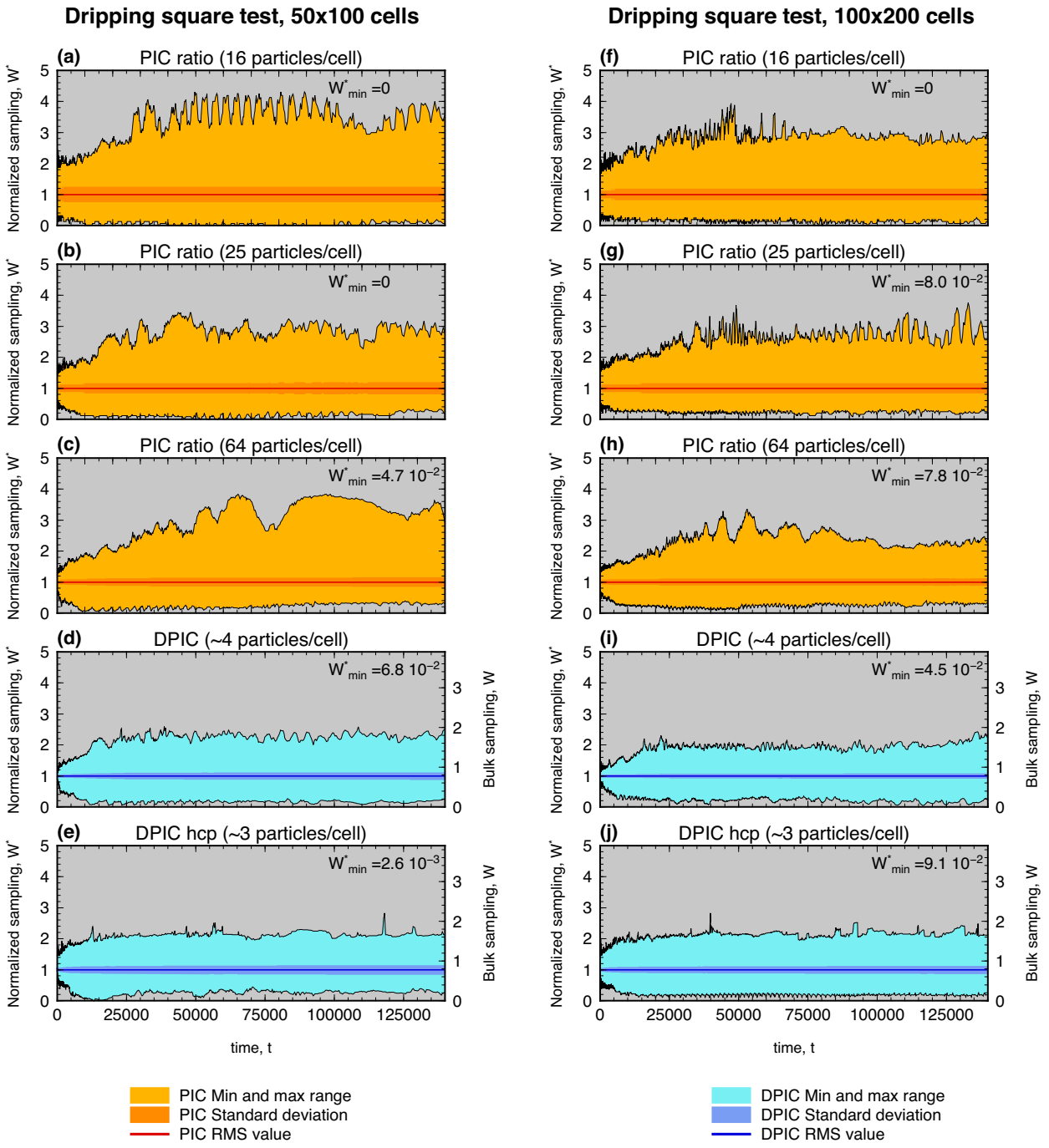

Figure 24. Results of the dripping instability test. Time evolution of the normalized cell sampling parameter, W^* corresponding to different initial values of particle numbers. The three top panels correspond to the PIC method using point-wise kernels, no particle remeshing, with 16, 25 and 64 particles per cell. The bottom panels correspond to the DPIC method with initially four particles per cell or three particles per cell using an initially more compact arrangement of the particle kernel (see Fig. 5d). The RMS value is represented by the thick curve, the min and max values are displayed by thin black lines and the standard deviation corresponds to the dark coloured area.

While it involves additional procedures, the DPIC method remains straightforward to implement. Further reduction of the computational cost and increase in accuracy can be expected by conducting a systematic investigation of the parameters considered for the DPIC (reduction of the calls to intrinsic functions, improvement of the spatial sampling thanks to the use of kernels of different size, optimization of the merging procedure...). These generalizations and improvements of the DPIC method will be the focus of future research.

However, even at this early stage, the DPIC method proves to be a very good alternative to standard PIC approaches: at comparable accuracy, the DPIC method can be four to six times more efficient than the PIC method.

Table 1. Distribution of the computational load for the different procedures involved in the DPIC method for the dripping instability test on a 100×200 grid. The procedure named 'kernel arrays' refers to various array assignments during RK time integration stages. 'Kernel update' refers to kernel time integration (eq. 20) together with other associated operations (computation of eigenvalues and eigenvector angles).

Procedure	Load (per cent)
Kernel to grid	35
Kernel merge	20
Kernel update	14
Particle advection	12
Kernel arrays	11
Kernel split	8

Table 2. Distribution of the computational load for the different procedures involved in the PIC method for the dripping instability test on a 100×200 grid. The procedure named 'kernel arrays' refers to various array assignments during RK time integration stages.

Procedure	Load (per cent)
Particle advection	54
Kernel arrays	33
Kernel to grid	13

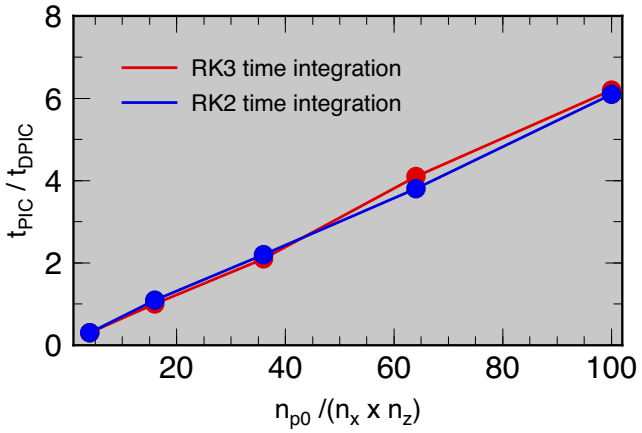

Figure 25. Performances of the PIC and DPIC methods. Execution time of the PIC method normalized to the execution time for the DPIC method performed using approximately four particles per cell. These correspond exclusively to the time associated with the resolution of eq. (1) for the dripping instability test over the first 1400 CFL time steps in a domain discretized using 100×200 square cells. Different amounts of particles are considered. Blue curves and symbols refer to the results obtained using a second-order TVD–RK time integration scheme. Red curves and symbols refer to the results obtained using a third-order TVD–RK time integration scheme.

ACKNOWLEDGEMENTS

I thank Jeroen van Hunen and an anonymous reviewer for their insightful comments, and the editor for his useful advice, which improved the quality of the manuscript. Discussions with J. van Hunen and R. Agrusta at an earlier stage of this work were also appreciated. I am also grateful to E. Ruhier for discussions on algorithmic aspects. This work has been supported by the Deutsche Forschugsgemeinschaft (project SA 2042/3), and by the INSU-CNES Programme National de Planétologie. Numerical computations were partly performed on the S-CAPAD platform, IPGP, France. All

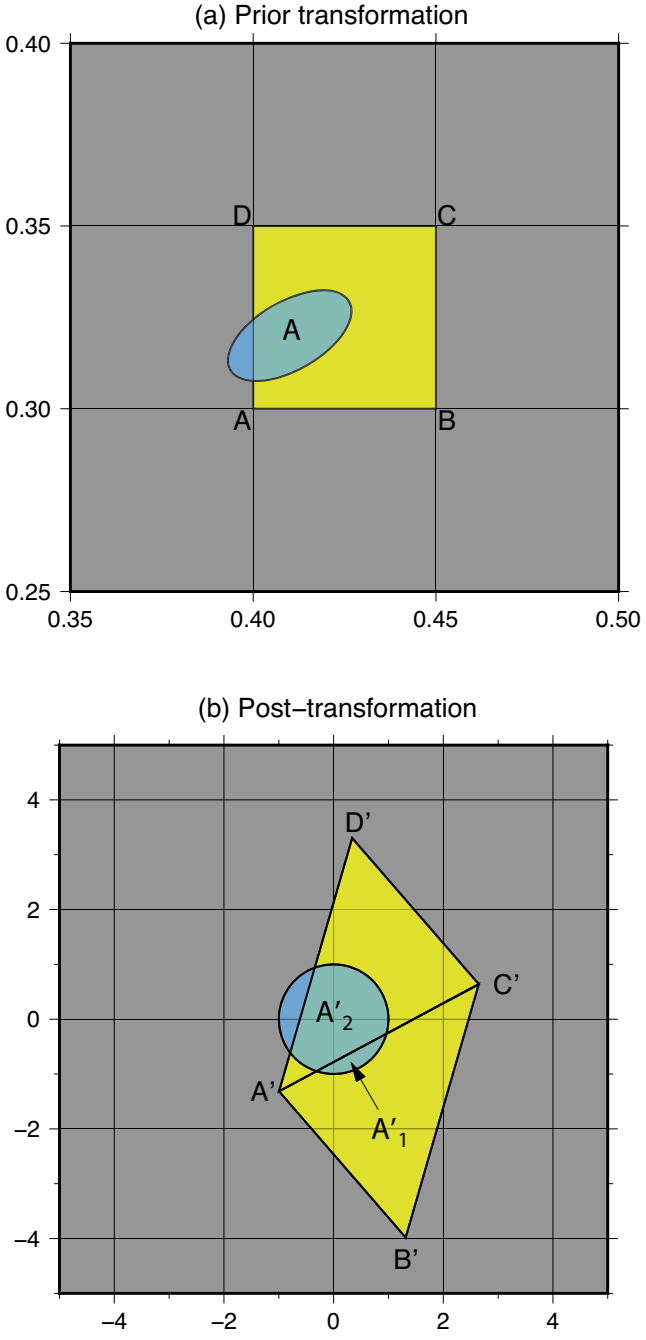

Figure 26. Illustration of the application of the transformation f_a (eq. B1) for the calculation of $\mathcal{A}$, the area common to a particle kernel (blue ellipse in the top panel) and the corresponding cell of size h^2 to which the particle belongs (yellow square in the top panel). In this example $r_p = h/4$, $h = 0.05$, $\sigma_p^+ = 1.5$, $\sigma_p^- = 0.75$, $x_p = 0.41$, $z_p = 0.32$ and $\alpha_p = \pi/6$. Upon application of f_a, A, B, C and D become A′, B′, C′ and D′ and the ellipse kernel is transformed into a unit circle, whose centre is located at $x' = z' = 0$ (bottom panel). A′, B′, C′ and D′ do not correspond to a square anymore but can be decomposed into two triangles. The sum of the common areas to the triangles A′B′C′ and A′C′D′, and the unit circle, that is, $\mathcal{A}'_1$ and $\mathcal{A}'_2$ can be calculated analytically by considering the number of triangle vertex belonging to the unit circle and the number of edges crossing or contained within the unit circle (Fig. 27), and by decomposing each common area into disc segments and triangles. See text for further details.

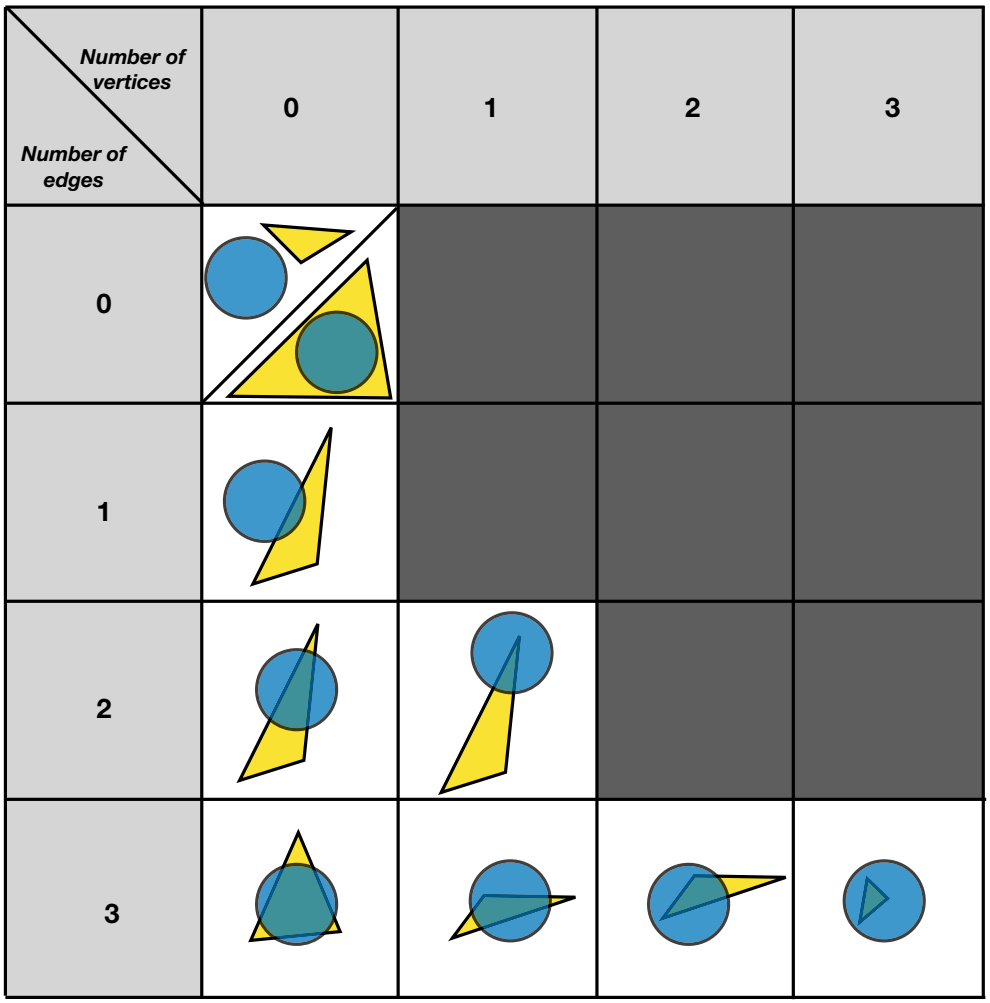

Figure 27. Possible overlapping configurations between the triangles resulting from the application of the transformation f_a (eq. B1, see Fig. 26), and the unit circle corresponding to the transformed particle kernel. The different cases depend on the number of triangle vertices present within the unit circle, and the number of triangle edges that partially or entirely belong to the unit circle. For each case, the overlapping area between the triangle and the unit circle can be decomposed into triangles and/or disc segments, whose surfaces can be computed analytically. See text for further details.

figures but Figs 1, 4–8, 22 and 27 were drawn with the Generic Mapping Tools (Wessel & Smith 1995). This is IPGP contribution number 3952.

REFERENCES

Bateson, W.B. & Hewett, D.W., 1998. Grid and particle hydrodynamics: beyond hydrodynamics via fluid element particle-in-cell, *J. Comput. Phys.,* **144,** 358–378.

Bouffard, M., Labrosse, S., Choblet, G., Fournier, A., Aubert, J. & Tackley, P.J., 2017. A particle-in-cell method for studying double-diffusive convection in the liquid layers of planetary interiors, *J. Comput. Phys.,* **346,** 552–571.

Brackbill, J.U., 2005. Particle methods, *Int. J. Numer. Methods Fluids,* **47,** 693–705.

Brackbill, J.U., Kothe, D.B. & Ruppel, H.M., 1987. Flip: a low-dissipation, particle-in-cell method for fluid flow, *Comput. Phys. Commun.,* **48**(1), 25–38.

Brunton, S.L. & Rowley, C.W., 2010, Fast computation of finite-time Lyapunov exponent fields for unsteady flows, *Chaos,* **20**.

Coppa, G.G., Lapenta, G., Dellapiana, G., Donato, F. & Riccardo, V., 1996. Blob method for kinetic plasma simulation with variable-size particles, *J. Comput. Phys.,* **127,** 268–284.

Deubelbeiss, Y. & Kaus, B.J., 2008. Comparison of Eulerian and Lagrangian numerical techniques for the Stokes equations in the presence of strongly varying viscosity, *Phys. Earth planet. Inter.,* **171,** 92–111.

Duretz, T., May, D.A., Gerya, T.V. & Tackley, P.J., 2011. Discretization errors and free surface stabilization in the finite difference and marker-in-cell method for applied geodynamics: a numerical study, *Geochem. Geophys. Geosyst.,* **12**(7),

Edwards, E. & Bridson, R., 2012. A high-order accurate particle-in-cell method, *Int. J. Numer. Methods Eng.,* **90**(9), 1073–1088.

Farnetani, C.G. & Samuel, H., 2003. Lagrangian structures and stirring in the Earth's mantle, *Earth planet. Sci. Lett.,* **206,** 335–348.

Fuchs, L. & Schmeling, H., 2013. A new numerical method to calculate inhomogeneous and time-dependent large deformation of two-dimensional geodynamic flows with application to diapirism, *Geophys. J. Int.,* **194,** 623–639.

Gerya, T.V., 2010. *Introduction to Numerical Geodynamic Modeling*. Cambridge University Press.

Gerya, T.V. & Yuen, D.A., 2003. Characteristics-based marker-in-cell method with conservative finite-differences schemes for modeling geological flows with strongly variable transport properties, *Phys. Earth planet. Inter.,* **140,** 293–318.

Gerya, T.V. & Yuen, D.A., 2007. Robust characteristics method for modeling multiphase visco-elasto-plastic thermo-mechanical problems, *Phys. Earth planet. Inter.,* **163,** 83–105.
Grigoryev, Y.N., Vshivkov, V.A. & Fedoruk, M.P., 2002. *Numerical "Particle-in-Cell" Methods: Theory and Applications,* VSP BV.
Harlow, F.H., 1957. Hydrodynamic problems involving large fluid distortions, *J. ACM,* **4**(2), 137–142.
Harlow, F.H., 1964. The particle-in-cell computing method for fluid dynamics, *Methods Comput. Phys.,* **3,** 319–343.
Hoïnk, T., Schmalzl, J. & Hansen, U., 2005. Formation of compositional structures by sedimentation in vigorous convection, *Phys. Earth planet. Inter.,* **153,** 11–20.
Kothe, D.B., 1998. Perspective on Eulerian finite volume methods for incompressible interfacial flows, eds ,Kuhlmann, H.C. & ,Rath, H.-J., in *Free Surface Flows,* Springer-Verlag, pp. 268–331.
Lapenta, G., 2012. Particle simulations of space weather, *J. Comput. Phys.,* **231,** 795–821.
Legras, B. & Dritschel, D.G., 1991. The elliptical model of two-dimensional vortex dynamics I: the basic state, *Phys. Fluids A,* **3,** 845–854.
Lin, J.-R., Gerya, T.V., Tackley, P.J., Yuen, D.A. & Golabek, G.J., 2011. Protocore destabilization in planetary embryos formed by cold accretion: feedbacks from non-Newtonian rheology and energy dissipation, *Icarus,* **213,** 24–42.
Maurice, M., Tosi, N., Samuel, H., Plesa, A.-C., Hüttig, C. & Breuer, D., 2017. Onset of solid-state mantle convection and mixing during magma ocean solidification, *J. geophys. Res.,* **122,** 577–598.
McKenzie, D.P., 1979. Finite deformation during fluid flow, *Geophys. J. R. astr. Soc.,* **58,** 689–715.
McNamara, A.K. & Zhong, S., 2005. Thermochemical structures beneath Africa and the Pacific ocean, *Nature,* **437,** 1136–1139.
Meyer, D.W. & Jenny, P., 2004. Conservative velocity interpolation for PDF methods, *Proc. Appl. Math. Mech.,* **4,** 466–467.
Monaghan, J.J., 1985. Particle methods for hydrodynamics, *Comput. Phys. Rep.,* **3,** 71–124.
Monaghan, J.J., 1992. Smoothed particle hydrodynamics, *Annu. Rev. Astron. Astrophys.,* **30,** 543–574.
Moresi, L., Dufour, F. & Mühlhaus, H.B., 2003. A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials, *J. Comput. Phys.,* **184**(2), 476–497.
Moresi, L., Quenette, S., Lemiale, V., Mériaux, C., Appelbe, B. & Mühlhaus, H.B., 2007. Computational approaches to studying non-linear of the crust and mantle, *Phys. Earth planet. Inter.,* **163,** 69–82.
Poliakov, A. & Podladchikov, Y., 1992. Diapirism and topography, *Geophys. J.,* **109,** 553–564.
Pusok, A.E., Kaus, B.J.P. & Popov, A.A., 2017. On the quality of velocity interpolation schemes for marker-in-cell method and staggered grids, *Pure appl. Geophys.,* **174,** 1071–1089.
Rider, W.J. & Kothe, D.B., 1995, A marker particle method for interface tracking . *Proceedings of the Sixth International Symposium on Computational Fluid Dynamics*. Springer, St Petersburg, Russia, pp. 1–7.
Ruprecht, P., Bergantz, G.W. & Dufek, J., 2008. Modeling of gas-driven magmatic overturn: tracking of phenocryst dispersal and gathering during magma mixing, *Geochem. Geophys. Geosyst.,* **9**.
Samuel, H., 2012a. A re-evaluation of metal diapir breakup and equilibration in terrestrial magma oceans, *Earth planet. Sci. Lett.,* **313–314,** 105–114.
Samuel, H., 2012b. Time-domain parallelization for computational geodynamics, *Geochem. Geophys. Geosyst.,* **13**.
Samuel, H. & Bercovici, D., 2006. Oscillating and stagnating plumes in the Earth's lower mantle, *Earth planet. Sci. Lett.,* **248,** 90–105.
Samuel, H. & Evonuk, M., 2010. Modeling advection in geophysical flows with particle level sets, *Geochem. Geophys. Geosyst.,* **11**(Q08020),
Samuel, H. & Farnetani, C.G., 2003. Thermochemical convection and helium concentrations in mantle plumes, *Earth planet. Sci. Lett.,* **207,** 39–56.
Shu, C.-W. & Osher, S., 1988. Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.,* **77,** 439–471.
Tackley, P.J., 2008. Modelling compressible mantle convection with large viscosity contrasts in a three-dimensional spherical shell using the yin-yang grid, *Phys. Earth planet. Inter.,* **171,** 7–18.
Tackley, P.J. & King, S.D., 2003. Testing the tracer ratio method for modeling active compositional fields in mantle convection simulations, *Geochem. Geophys. Geosyst.,* **4**(4)
Thielmann, M., May, D.A. & Kaus, B.J.P., 2014. Discretization errors in the hybrid finite element particle-in-cell method, *Pure appl. Geophys.,* **171,** 2165–2184.
Tosi, N. *et al.* 2015. A community benchmark for viscoplastic thermal convection in a 2-D squared box. *Geochem. Geophys. Geosyst.,* **16,** 2175–2196.
van Hunen, J., van den Berg, A.P. & Vlaar, N.J., 2004. Various mechanisms to induce present-day shallow flat subduction and implications for the younger Earth: a numerical parameter study, *Phys. Earth planet. Inter.,* **146,** 179–174.
van Keken, P.E., King, S.D., Schmeling, H., Christensen, U.R., Neumeister, D. & Doin, M.-P., 1997. A comparison of methods for the modeling of thermochemical convection, *J. geophys. Res.,* **102,** 22 477–22 495.
Wang, H., Agrusta, R. & Van Hunen, J., 2015. Advantages of a conservative velocity interpolation (CVI) scheme for particle-in-cell methods with application in geodynamic modeling, *Geochem. Geophys. Geosyst.,* **16**(6), 2015–2023.
Welling, U. & Germano, G., 2011. Efficiency of linked cell algorithms, *Comput. Phys. Commun.,* **182**(3), 611–615.
Wessel, P. & Smith, W.H.F., 1995. A new version of the Generic Mapping Tools (GMT), *EOS, Trans. Am. geophys. Un.,* **76,** 329, doi:10.1029/95EO00198.
Zhong, S.J., 1996. Analytic solutions for stokes' flow with lateral variations in viscosity, *Geophys. J. Int.,* **124,** 18–28.

APPENDIX A: ANALYTICAL EXPRESSIONS FOR THE PARTIAL SAMPLING FOR THE PIC METHOD FOR THE VORTEX TEST

Here I derive the analytical expression for the distance between 'corner' particles, located in the vicinity of $x = z = 0$ (see Fig. 3c) advected in the case of a vortex flow (eq. 14). The unit square domain is discretized using $n \times n$ square cells of size $h = 1/n$. The particles are initially regularly spaced within the domain. Taylor expansion of the velocity field around $x = z = 0$ yields

$$\mathbf{u}(x = 0, z = 0) \cong (\pi x, -\pi z)^T. \tag{A1}$$

Namely, the flow in this region corresponds essentially to a stagnation point associated with pure shear. Consider the two particles, C and D, initially located at $x_{C_0} = 0.25h, z_{C_0} = 0.75h$ and $x_{D_0} = z_{C_0}$, $z_{D_0} = x_{C_0}$, such that C and D are initially close to each other, and they belong to the same streamline. While these two particles follow the same trajectory, the velocity along the corresponding streamline varies strongly in this region, and so does l_{CD}, the distance between particles C and D. Indeed, integrating eq. (2) with $\mathbf{u}_p = \mathbf{u}(x = 0, z = 0)$ and using eq. (A1) yields the approximate location of particles in the vicinity of $(x = 0, z = 0)$:

$$\mathbf{x}_p \cong (x_0 \exp(\pi t), z_0 \exp(-\pi t)). \tag{A2}$$

Using the above equation, one can express the distance between particles C and D:

$$l_{\mathrm{CD}} \cong 0.5h\sqrt{\exp(2\pi t) + \exp(-2\pi t)}). \tag{A3}$$

Recognizing that the second term with the decreasing exponential is bounded between 0 and 1, and will not significantly contribute to l_{CD}, we have

$$l_{\mathrm{CD}} > h \exp(\pi t)/2. \tag{A4}$$

APPENDIX B: ANALYTICAL COMPUTATION OF THE PARTICLE'S WEIGHTS FOR THE DPIC METHOD

The elliptical kernels in the DPIC method do not allow a direct analytical computation of their overlapping areas with grid cells. For this reason, a transformation, f_a, is first applied to each of the four cell corners a particle belongs to. This mapping consists of a translation, a rotation and shrinking/expansion that converts the elliptic kernel of a given particle into a disc of unit radius centred on the particle's position $\mathbf{x}_p$:

$$f_a(\mathbf{x}) = \mathbf{R}(\mathbf{x} - \mathbf{x}_p)\mathbf{D}, \tag{B1}$$

where $\mathbf{R}$ is a rotation matrix

$$\mathbf{R} = \begin{pmatrix} \cos\alpha_p & \sin\alpha_p \\ -\sin\alpha_p & \cos\alpha_p \end{pmatrix}, \tag{B2}$$

and $\mathbf{D}$ is a deformation matrix given by

$$\mathbf{D} = \begin{pmatrix} 1/(r_p\sigma_p^+) & 0 \\ 0 & 1/(r_p\,\sigma_p^-) \end{pmatrix}. \tag{B3}$$

The mapping applied to the four cell corners A, B, C and D results in two triangles A′B′C′ and A′C′D′ (see Fig. 26). The area overlap between the unit disc and each of these two triangles are $\mathcal{A}'_1$ and $\mathcal{A}'_2$. These areas can be evaluated exactly by distinguishing between the nine possible cases depending on the number of triangle vertices present within the unit circle, and the number of triangle edges that partially or entirely belong to the unit circle (Fig. 27). For each case, the overlapping area between A′B′C′ and A′C′D′ and the unit circle is then decomposed into triangles and/or disc segments, whose surfaces are computed analytically. Finally, with the knowledge of $\mathcal{A}'_1$ and $\mathcal{A}'_2$, the overlapping area between the ellipse and the cell is deduced by applying the inverse f_a mapping, yielding $(\mathcal{A}'_1 + \mathcal{A}'_2)r_p^2\sigma_p^+\sigma_p^-$.